OPTIMIERUNG LINEARER SYSTEME MIT ZUSÄTZLICHEN DISJUNKTIVITÄTSBEDINGUNGEN

D I S S E R T A T I O N

zur Erlangung des Grades eines
Doktors der Naturwissenschaften
der Abteilung für Mathematik
der Ruhr-Universität Bochum

von

Hans Weinberger

Bochum 1981

OPTIMIERUNG LINEARER SYSTEME
MIT ZUSÄTZLICHEN DISJUNKTIVITÄTSBEDINGUNGEN

D I S S E R T A T I O N

zur Erlangung des Grades eines
Doktors der Naturwissenschaften
der Abteilung für Mathematik
der Ruhr-Universität Bochum

von

Hans Weinberger

Bochum 1981

UNIV.
LEIDEN
BIBL.

Inhaltsverzeichnis

		Seite
	Vorbemerkung	3
	Bezeichnungen	4
1.	Einführung	5
2.	Anwendungen	11
3.	Simplexverfahren	21
4.	Erweitertes Simplexverfahren	35
5.	Aus der ganzzahligen Optimierung abgeleitetes allgemeines Lösungsverfahren	54
6.	Schnittebenenverfahren	59
7.	Branch-and-Bound-Verfahren	67
8.	Kombination von Schnittebenenverfahren und Branch-and-Bound-Verfahren	73
9.	Effizientes Lösungsverfahren	74
10.	Implementierung und Rechenerfahrung	86
	Schlußbemerkung	90
	Literaturverzeichnis	91

Vorbemerkung

Nach den bahnbrechenden Erfolgen des Simplexverfahrens in der linearen Optimierung lag die Vermutung nahe, daß auch innerhalb der nichtlinearen Optimierung zumindest auf gewissen Teilbereichen ähnlich effiziente Lösungsverfahren entwickelbar sind,d.h. Verfahren, die es ermöglichen, für große Probleme, wie sie in der Praxis auftreten, in wirtschaftlich vertretbarer Rechenzeit eine hinreichend gute Lösung zu finden.
Leider konnte dies bisher nicht erreicht werden.
Hierbei einen Fortschritt zu schaffen, ist das Ziel der vorliegenden Arbeit.
Dazu wird vorgeschlagen, nichtlineare Optimierungssysteme - wenn möglich - in lineare Systeme mit zusätzlichen speziellen nichtlinearen Restriktionen, die Disjunktivitätsbedingungen genannt werden, aufzuspalten (§1 und §2).
Eine solche Formulierung ermöglicht es, das Simplexverfahren in naheliegender Weise zu erweitern (§3 und §4) und auf besonders günstige Art mit Lösungsprinzipien aus der ganzzahligen Optimierung (§5 - §8) zu kombinieren.
Die Effizienz des daraus resultierenden Verfahrens (§9) beruht darauf, daß durch die Verwendung vorgeschalteter heuristischer Strategien sowie sukzessiver Verkleinerung des Lösungsbereiches mittels Einführung von Schnittebenen und verschärften Anspruchsniveaus der Suchaufwand für die sich anschließenden endlichen Methoden erheblich reduziert wird. Da das Verfahren sich stark an die Simplexmethode anlehnt, läßt es sich innerhalb eines modernen, leistungsfähigen Softwaresystems implementieren (§ 10). Schnelligkeit und Lösungsqualität sind dabei vom Benutzer steuerbar.
Die vorliegende Arbeit hat somit das Ziel, einen Beitrag zur mathematischen Optimierung zu leisten, der den Bedürfnissen der Praxis gerecht wird.

Bezeichnungen

$A = ({}_mA_n)_{(m,n)\in MxN}$: (MxN)-Matrix

${}_RA_P$: (RxP)-Untermatrix von A

${}_RA := {}_RA_N$; $A_P := {}_MA_P$

${}_mA$: Zeile m von A

A_n : Spalte n von A

$x = (x_n)_{n\in N}$: N-Tupel

x_P : P-Untertupel von x

$x \leqq y :\Leftrightarrow x_i \leqq y_i \quad \forall i$

$|x| = (|x_n|)_{n\in N}$; x: N-Tupel

$A\cdot x = Ax = \left(\sum_{n\in N} {}_mA_n \cdot x_n\right)_{m\in M}$; A : (MxN)-Matrix; x : N-Tupel

$y\cdot A = yA = \left(\sum_{m\in M} {}_mA_n \cdot y_m\right)_{n\in N}$; A : (MxN)-Matrix; y : M-Tupel

$x\cdot y = xy = \sum_{n\in N} x_n \cdot y_n$; x,y : N-Tupel

(Es wird also hier nicht zwischen Zeilentupel und Spaltentupel unterschieden.)

$|M|$: Kardinalzahl der Menge M

$a + \infty = + \infty + a = + \infty \quad \forall \mathbb{R} \cup \{+\infty\}$

$a - \infty = - \infty + a = - \infty \quad \forall \mathbb{R} \cup \{-\infty\}$

1. Einführung

Gegeben sei das folgende Optimierungsproblem:

$$\underline{c}\underline{x} \rightarrow \max! \tag{1}$$

$$\underline{a}^- \leqq \underline{A}\underline{x} \leqq \underline{a}^+ \tag{2}$$

$$\underline{b}^- \leqq \underline{x} \leqq \underline{b}^+ \tag{3}$$

$$\prod_{i\in\underline{M}_k^-}({}_i\underline{A}\underline{x}-\underline{a}_i^-)\cdot\prod_{i\in\underline{M}_k^+}(\underline{a}_i^+-{}_i\underline{A}\underline{x})\cdot\prod_{j\in\underline{N}_k^-}(\underline{x}_j-\underline{b}_j^-)\cdot\prod_{j\in\underline{N}_k^+}(\underline{b}_j^+-\underline{x}_j)=0 \text{ für alle } k\in\underline{K} \tag{4}$$

mit $\underline{A}$: $(\underline{M}\times\underline{N})$-Matrix,

$\underline{c}$: reelles $\underline{N}$-Tupel,

$\underline{b}^-$: $\underline{N}$-Tupel, dessen Komponenten reell oder $-\infty$ sind,

$\underline{b}^+$: $\underline{N}$-Tupel, dessen Komponenten reell oder $+\infty$ sind,

$\underline{a}^-$: $\underline{M}$-Tupel, dessen Komponenten reell oder $-\infty$ sind,

$\underline{a}^+$: $\underline{M}$-Tupel, dessen Komponenten reell oder $+\infty$ sind.

($_i\underline{A}$ bezeichnet dabei die Zeile i von $\underline{A}$, $\underline{a}_i^-$ die Komponente i von $\underline{a}^-$ etc.)

Ferner seien $\underline{M}$, $\underline{N}$ und $\underline{K}$ endliche Indexfamilien und

$\underline{M}_k^-, \underline{M}_k^+ \subseteq \underline{M}$; $\underline{N}_k^-, \underline{N}_k^+ \subseteq \underline{N}$ für $k\in\underline{K}$.

Die Restriktionen (4) können als Disjunktionen interpretiert werden. Gesucht werden $\underline{N}$-Tupel $\underline{x}$, die außer den linearen Restriktionen (2), (3) für jedes $k\in\underline{K}$ mindestens eine der Randbedingungen.

$$\begin{aligned} \underline{a}_i^- &= {}_i\underline{A}\,\underline{x} && i\in\underline{M}_k^- \\ {}_i\underline{A}\,\underline{x} &= \underline{a}_i^+ && i\in\underline{M}_k^+ \\ \underline{b}_j^- &= \underline{x}_j && j\in\underline{N}_k^- \\ \underline{x}_j &= \underline{b}_j^+ && j\in\underline{N}_k^+ \end{aligned}$$

erfüllen.

Für die rechentechnische Behandlung ist es vorteilhaft, das Ausgangsproblem (1) - (4) durch Einführung von logischen Variablen und durch Koordinatentransformationen auf die folgende Gestalt zu bringen:

$$cx \to \max! \qquad (\underset{\sim}{1})$$

$$Ax = a \qquad (\underset{\sim}{2})$$

$$b^- \leqq x \leqq b^+ \qquad (\underset{\sim}{3})$$

$$\prod_{j\in N_k^-} x_j \cdot \prod_{j\in N_k^+} (b_j^+ - x_j) = 0 \quad \text{für alle } k\in K\ . \qquad (\underset{\sim}{4})$$

Dabei treten dann nur noch Variablen der folgenden Typen auf:

$$0 \leqq x_j \leqq b_j^+ \qquad \text{(Typ 1)}$$

$$0 \leqq x_j \leqq +\infty \qquad \text{(Typ 2)}$$

$$-\infty \leqq x_j \leqq 0 \qquad \text{(Typ 2')}$$

$$-\infty \leqq x_j \leqq +\infty \qquad \text{(Typ 3)}$$

und ($\underline{4}$) erhält die vereinfachte Gestalt ($\underset{\sim}{4}$).
Die Beziehung zum Ausgangsproblem ($\underline{1}$) - ($\underline{4}$) wird durch die folgenden Formeln hergestellt: [1)]

$$\underline{y} = \underline{A}\,\underline{x}\ ,$$

$$l = \begin{pmatrix} \underline{b}^- \\ \underline{a}^- \end{pmatrix}\ ,\quad L = \begin{pmatrix} \underline{b}^+ \\ \underline{a}^+ \end{pmatrix}\ ,\quad y = \begin{pmatrix} \underline{x} \\ \underline{y} \end{pmatrix}\ ,$$

$$R_i = \begin{cases} l_i, \text{ falls } l_i \neq -\infty \\ L_i, \text{ falls } l_i = -\infty \text{ und } L_i \neq +\infty \\ 0\ , \text{ falls } l_i = -\infty \text{ und } L_i = +\infty \end{cases}$$

$x = y\text{-}R$, $A = (\underline{A}\,|\,-E)$, $c = (\underline{c}\,|\,0)$, $a = -AR$, $b^- = l\text{-}R$, $b^+ = L\text{-}R$, $N_k^- = \underline{N}_k^- \cup \underline{M}_k^-$, $N_k^+ = \underline{N}_k^+ \cup \underline{M}_k^+$, $K = \underline{K}$, $M = \underline{M}$, $N = \underline{N} \cup \underline{M}$.

---

1) Vergl. [Ciriani, 1972], [Orchard-Hays, 1968].

Dabei ist

A : (MxN)-Matrix,
E : (MxM)-Einheitsmatrix,
a : M-Tupel,
O : M-Nulltupel,
c,l,L,R,b^-,b^+: N-Tupel.

Ersetzt man nun noch in (1) - (4) die Variablen x_j vom (Typ 2') durch $-x_j$, so erhält man die Normalform (1) - (4), in der nur drei Typen (Typ 1, Typ 2, Typ 3) von Variablen auftreten und von der aus die Optimierung beginnt.

Beispiel: Als Ausgangsproblem sei gegeben:

$$
\begin{array}{rcl}
 & \underline{x}_1 - 7\underline{x}_2 + 3\underline{x}_3 + 5\underline{x}_4 \rightarrow \max! & \\
0 \leqq & \underline{x}_1 + \underline{x}_4 & \leqq 7 \\
1 \leqq & -\underline{x}_1 + \underline{x}_3 & \leqq 3 \\
-1 \leqq & \underline{x}_1 - \underline{x}_2 & \leqq 2 \\
-\infty \leqq & \underline{x}_2 - \underline{x}_3 + \underline{x}_4 & \leqq 1 \\
\hline
0 \leqq & \underline{x}_1 & \leqq 2 \\
-\infty \leqq & \underline{x}_2 & \leqq +\infty \\
-4 \leqq & \underline{x}_3 & \leqq 4 \\
0 \leqq & \underline{x}_4 & \leqq +\infty \\
 & (7-\underline{x}_1-\underline{x}_4)\cdot\underline{x}_4 & = 0 \\
 & (2-\underline{x}_1)\cdot(\underline{x}_3+4) & = 0 \\
 & \underline{x}_1\cdot(\underline{x}_3+4)\cdot(4-\underline{x}_3) & = 0 \\
 & (-\underline{x}_1+\underline{x}_3-1)\cdot(2-\underline{x}_1+\underline{x}_2)\cdot\underline{x}_4 & = 0
\end{array}
$$

Es ist also

$$
l = \begin{pmatrix} 0 \\ -\infty \\ -4 \\ 0 \\ 0 \\ 1 \\ -1 \\ -\infty \end{pmatrix} \qquad L = \begin{pmatrix} 2 \\ +\infty \\ 4 \\ +\infty \\ 7 \\ 3 \\ 2 \\ 1 \end{pmatrix} \text{ und somit } R = \begin{pmatrix} 0 \\ 0 \\ -4 \\ 0 \\ 0 \\ 1 \\ -1 \\ 1 \end{pmatrix}
$$

Daraus erhält man

$$b^- = \begin{pmatrix} 0 \\ -\infty \\ 0 \\ 0 \\ 0 \\ 0 \\ 0 \\ -\infty \end{pmatrix}, \quad b^+ = \begin{pmatrix} 2 \\ +\infty \\ 8 \\ +\infty \\ 7 \\ 2 \\ 3 \\ 0 \end{pmatrix} \quad \text{und} \quad a = \begin{pmatrix} 0 \\ 5 \\ -1 \\ -3 \end{pmatrix}.$$

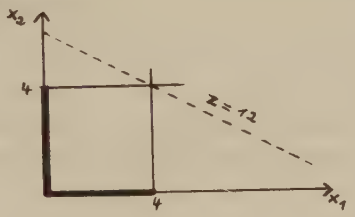
Ferner sind zur Umformung der disjunktiven Restriktionen die Beziehungen $x_1 = \underline{=} x_1$, $x_3 = \underline{=} x_3 + 4$, $x_4 = \underline{=} x_4$, $x_5 = \underline{=} x_1 + \underline{=} x_4$, $x_6 = \underline{=} {-x_1} + \underline{=} x_3 - 1$, $x_7 = \underline{=} x_1 - \underline{=} x_2 + 1$ zu beachten. Nach der Ersetzung von x_8 durch $-x_8$ ergibt sich damit als Normalform für das Ausgangsproblem

$$\begin{array}{rrrr

Das lineare System (1) - (3) hat bekanntlich [1] die folgenden, für die numerische Behandlung überaus günstigen Eigenschaften:

(a) Die Menge der zulässigen Lösungen ist konvex.
(b) Ein lokales Optimum ist zugleich auch globales Optimum.
(c) Existiert ein Optimum, dann existiert auch eine optimale Basislösung.

Offensichtlich hat das System (1) - (4) die Eigenschaften (a) und (b) i.a. nicht mehr, wie beispielsweise die geometrische Interpretation



des Systems

$$z = x_1 + 2x_2 \rightarrow \max!$$
$$0 \leqq x_1 \leqq 4$$
$$0 \leqq x_2 \leqq 4$$
$$x_1 \cdot x_2 = 0$$

unmittelbar zeigt. Die Eigenschaft (c) ist jedoch weiterhin gültig:

Satz: Besitzt das System (1) - (4) eine optimale Lösung, so besitzt es auch eine optimale Basislösung.

Dabei versteht man als naheliegende Erweiterung der für Linearprogramme bekannten Definitionen unter einer Lösung bzw. insbesondere Basislösung von (1) - (4) eine Lösung bzw. Basislösung von (1) - (3), die zusätzlich (4) erfüllt.

---

1) Siehe z.B. [Neumann, 1975, S. 25 ff.] oder [Kall, 1976, S. 16 ff].

Beweis des Satzes: Theoretisch läßt sich eine optimale Lösung von (1) - (4) wie folgt bestimmen: Man fixiert eine Anzahl von Variablen so auf Null bzw. auf ihre oberen Grenzen, daß (4) erfüllt ist, und löst das daraus resultierende Linearprogramm mit Hilfe der Simplexmethode. Dies läßt sich nur auf endlich viele verschiedene Weisen durchführen. Aus der so erhaltenen Menge von zulässigen Lösungen von (1) - (4) wählt man diejenige mit dem größten Zielwert. Der Satz ergibt sich damit unmittelbar aus der Gültigkeit von (c) für (1) - (3).

□

2. Anwendungen

In diesem Paragraphen sollen einige Probleme aufgezählt werden, die sich auf die Form (1) - (4) zurückführen lassen.

2.1 Gemischt-ganzzahlige 0/1-Optimierungsprobleme:

$$(c^1 | c^2) \begin{pmatrix} x \\ y \end{pmatrix} \to \max!$$

$$(A^1 | A^2) \begin{pmatrix} x \\ y \end{pmatrix} = a$$

$$b^- \leqq x \leqq b^+$$

$$y \in \{0,1\}^{N''} \quad ,$$

wobei y ein N"-Tupel mit Komponenten aus {0,1} sei und $x \in \mathbb{R}^{N'}$.

Das obige System kann man auf das folgende lineare System mit zusätzlichen Disjunktivitätsbedingungen zurückführen:

$$(c^1 | c^2) \begin{pmatrix} x \\ y \end{pmatrix} \to \max!$$

$$(A^1 | A^2) \begin{pmatrix} x \\ y \end{pmatrix} = a$$

$$\begin{pmatrix} b^- \\ 0 \end{pmatrix} \leqq \begin{pmatrix} x \\ y \end{pmatrix} \leqq \begin{pmatrix} b^+ \\ 1 \end{pmatrix}$$

$$y \cdot (1-y) = 0 \; .$$

Hierbei ist $1 \in \mathbb{R}^{N''}$ das konstante Einstupel.

2.2 Quadratische Optimierungsprobleme:

$$\begin{aligned} cx + \tfrac{1}{2}xCx &\to \max! \\ Ax &= a \\ x &\geqq 0 \,, \end{aligned} \qquad (5)$$

wobei c ein N-Tupel und C eine reelle symmetrische (NxN)-Matrix sei. [1]

Ist x^o ein lokales Maximum des obigen Systems, dann existieren u^o, v^o, so daß die Kuhn-Tucker-Bedingungen

$$\begin{aligned} Ax &= a \\ v + Cx - A^T u &= -c \\ x, v &\geqq 0 \\ x \cdot v &= 0 \end{aligned} \qquad (6)$$

erfüllt sind. [2] Daraus erhält man:

$$\begin{aligned} cx^o + \tfrac{1}{2}x^oCx^o &= cx^o + \tfrac{1}{2}(-cx^o + u^oAx^o - v^ox^o) \\ &= \tfrac{1}{2}(cx^o + u^oa) \end{aligned}$$

und somit ein globales Maximum von (5), wenn man die Zielfunktion

$$\tfrac{1}{2}(cx + au)$$

unter den Restriktionen (6) maximiert.

2.3 Lineare Komplementärprobleme:

$$\begin{aligned} w &= By + q \\ y \geqq 0,\ w &\geqq 0 \\ w \cdot y &= 0, \end{aligned}$$

wobei B eine reelle (MxM)-Matrix und q ein reelles M-Tupel

---

1) Man beachte, daß C sonst keiner weiteren Einschränkung unterliegt.

2) Siehe z.B. [Hadley, 1969, S. 261]

sei, sind offensichtlich ein Spezialfall (ohne Zielfunktion) von (1) - (4).

Die drei bisher genannten Problemtypen spielen in der Praxis des Operations Research eine große Rolle. Sehr viele nichtlineare betriebliche Entscheidungsprobleme lassen sich durch Einführung von binären Variablen auf gemischt-ganzzahlige O/1-Optimierungsprobleme zurückführen, [1] z.B. Probleme

- mit sprungfixen Komponenten in der Zielfunktion und in den Restriktionen,
- mit Prioritätsregeln (Reihenfolgebedingungen),
- mit konditionalen Bedingungen, d.h. logischen Verknüpfungen von Restriktionen.

Betriebswirtschaftliche Entscheidungstatbestände, die zu (nichtkonkaven) quadratischen Optimierungsproblemen führen, treten sowohl in den Bereichen der Beschaffung und Produktion als auch des Absatzes und der Finanzierung auf. [2]

Zu linearen Komplementärproblemen gelangt man beispielsweise bei Anwendung der ökonomischen Gleichgewichtstheorie, der Portfolio-Selectionstheorie und der Oligopoltheorie. [3]

Vom praktischen Standpunkt aus gesehen, wäre es daher interessant zu prüfen, ob bzw. in welchen Fällen ein gemischt-ganzzahliges O/1-Optimierungsproblem, ein quadratisches Optimierungsproblem oder ein lineares Komplementärproblem nach

---

1) Siehe [IBM, Juli 1975a], [Brauer, 1969, Kapitel 1 und 2].

2) Siehe [Rammé, 1975, Teil V].

3) Siehe [Bastian, 1976, Kapitel 4].

Transformation in die Form (1) - (4) effizienter, insbesondere in Bezug auf Rechenzeit, gelöst werden kann. Darüber hinaus stellt sich die Frage, ob dann der Umweg über einen der drei obengenannten Problemtypen bei Aufstellung des mathematischen Modells überhaupt notwendig ist. Diese Frage ist in Bezug auf den erforderlichen Speicherplatz von Bedeutung, da zusätzliche Transformationen meistens zu einer erheblichen Vergrößerung der Arbeitsmatrix führen. Tatsächlich lassen sich in vielen Fällen praktische Entscheidungsprobleme am direktesten und elegantesten als lineare Systeme mit zusätzlichen Disjunktivitätsbedingungen formulieren. Hierfür einige Beispiele:

2.4 Lineare Optimierungsprobleme mit zusätzlichen "entweder - oder" Bedingungen:

Lineare Programme behandeln nur "und" Bedingungen, d.h. Situationen, in denen alle gegebenen Bedingungen erfüllt sein müssen. Zusätzlich auftretende "entweder - oder" Bedingungen führen unmittelbar zu einem System der Form (1) - (4). Kann beispielsweise die Belieferung eines Kunden von zwei Lägern erfolgen, von denen jedoch nur eines ausgewählt werden darf, so läßt sich diese Forderung am einfachsten durch die disjunktive Restriktion

$$x_1 \cdot x_2 = 0$$

ausdrücken, wobei x_i (i = 1,2) die vom Lager i ausgelieferte Menge bedeutet. Die Beschreibung desselben Sachverhalts mit Hilfe von 0/1-Variablen ist wesentlich umständlicher:

$$\begin{aligned} x_1 - b_1^+ y_1 &\leqq 0 \\ x_2 - b_2^+ y_2 &\leqq 0 \\ y_1 + y_2 &\leqq 1 \\ y_1, y_2 &\in \{0,1\}, \end{aligned}$$

wobei vorausgesetzt werden muß, daß x_i (i = 1,2) eine Variable vom Typ 1 ist, also nach oben durch b_i^+ (i = 1,2) beschränkt ist.

2.5 Lineare Optimierungsprobleme mit zusätzlichen Reihenfolgebedingungen:

Kann eine Aktivität erst dann einsetzen, wenn eine andere bereits abgeschlossen ist, dann ergeben sich Prioritätsregeln zwischen gewissen Variablen, z.B.: x_2 kann erst größer Null werden, wenn x_1 seine obere Grenze b_1^+ erreicht hat, und x_3 kann erst größer Null werden, wenn x_2 seine obere Grenze b_2^+ erreicht hat. Mit Hilfe von Restriktionen der Form (4) läßt sich diese Reihenfolge leicht erzwingen:

$$x_2 \cdot (b_1^+ - x_1) = 0$$
$$x_3 \cdot (b_2^+ - x_2) = 0 \ .$$

Zum Vergleich sei wiederum auch der gemischt-ganzzahlige Ansatz aufgeführt:

$$x_1 - b_1^+ y_1 \geqq 0$$
$$x_2 - b_2^+ y_2 \geqq 0$$
$$x_2 - b_2^+ y_1 \leqq 0$$
$$x_3 - b_3^+ y_2 \leqq 0$$
$$y_1, y_2 \in \{0,1\}.$$

In beiden Fällen sind x_1, x_2 und x_3 Variablen vom Typ 1.

2.6 Lineare Optimierungsprobleme mit zusätzlicher kardinaler Beschränktheitsbedingung:

Können beispielsweise für eine Mischung drei Bestandteile verwendet werden, von denen nur zwei ausgewählt werden dürfen, so läßt sich diese Forderung am einfachsten durch die disjunktive Restriktion

$$x_1 \cdot x_2 \cdot x_3 = 0$$

ausdrücken, wobei x_i $(i = 1,2,3)$ die vom Bestandteil i verwendete Menge bedeutet.

Der O/1-Ansatz führt zu den Restriktionen

$$x_1 - b_1^+ y_1 \leqq 0$$
$$x_2 - b_2^+ y_2 \leqq 0$$
$$x_3 - b_3^+ y_3 \leqq 0$$
$$y_1 + y_2 + y_3 \leqq 2$$
$$y_1, y_2, y_3 \in \{0,1\},$$

wobei x_i $(i = 1,2,3)$ Variablen vom Typ 1 seien.

Allgemeiner wären also zur Umschreibung der Bedingung $|\{x_j > 0 | j \in J\}| \leqq q$ mit O/1-Variablen, wobei $J \subseteq N$ (Indexfamilie der im linearen Optimierungsproblem auftretenden Variablen) fest vorgegeben sei, die $|J|+1$ zusätzlichen Restriktionen

$$x_j - b_j^+ y_j \leqq 0 , \qquad j \in J$$

$$\sum_{j \in J} y_j \leqq q$$

und die $|J|$ zusätzlichen O/1-Variablen $y_j (j \in J)$ erforderlich, was bei großem J zu Problemen führt, die unmöglich innerhalb einer wirtschaftlich vertretbaren Rechenzeit gelöst werden können.

Bei dem disjunktiven Ansatz

$$\prod_{j \in N_k^-} x_j = 0 \quad \forall \; N_k^- \subseteq J \text{ mit } |N_k^-| > q$$

hingegen treten diese Schwierigkeiten nicht auf, da die Mengen N_k^- während der Rechnung nie spezifiziert zu werden brauchen (siehe Seite 85).

Die eben genannten Beispiele belegen, daß sich viele betriebliche Entscheidungsprobleme am elegantesten in der Form (1) - (4) mathematisieren lassen.

Ein weiterer, für den Praktiker interessanter Aspekt ist, daß der disjunktive Ansatz neue Möglichkeiten schafft, nichtlineare Probleme angenähert zu lösen. Als Beispiel seien erwähnt:

2.7 Minimierungsprobleme [1] mit separierbarer konkaver Zielfunktion und linearen Restriktionen:

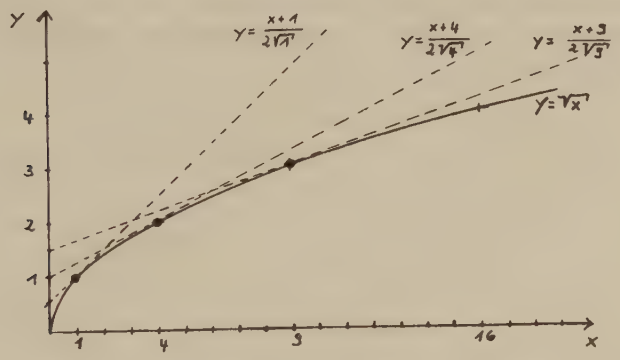
Zur Illustration sei angenommen, daß die mit einer Typ 2-Variablen x verbundenen Kosten

$$t(x) = \sqrt{x}$$

sind. Analytisch kann der Graph $y = \sqrt{x}$ als Enveloppe der Tangenten mit der Gleichung

$$y = \frac{x+p}{2\sqrt{p}},$$

definiert werden, wobei p ein nichtnegativer Parameter ist. [2]



1) Maximierungsprobleme sind hierbei uninteressant, da dann ein globales Optimum auch mit Hilfe linearer Approximation angenähert ermittelt werden kann, siehe z.B. [Henn/Künzi, 1968, S. 32 ff].

2) Siehe [Owen, 1973], [Courant, 1963, S. 149 ff.].

Wegen der Konkavität liegt diese Enveloppe unterhalb der Tangenten, so daß gilt:

$$t(x) = \min_{p \geq 0} \frac{x+p}{2\sqrt{p}}$$

Setzt man

$$u_p = -2\sqrt{p}\, t + x + p$$

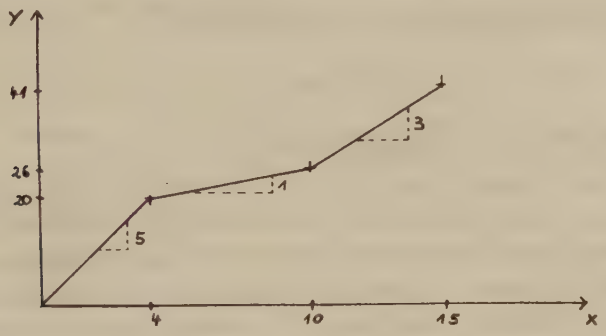
dann gilt (in Abhängigkeit von x) $u_p \geq 0$ für alle $p \geq 0$, jedoch $u_p = 0$ für mindestens ein p, nämlich für dasjenige p, für welches das obige Minimum angenommen wird. Man erhält auf diese Weise unendlich viele Variablen u_p, von denen mindestens eine verschwinden muß. Bei der numerischen Durchführung wählt man natürlich nur eine endlich Anzahl n von Werten für p. Man ersetzt also in der Zielfunktion $\sqrt{x}$ durch die Variable t und fügt zum Ausgangsproblem die Restriktionen

$$\begin{aligned} u_1 + 2\sqrt{1}\, t - x &= 1 \\ &\vdots \\ u_n + 2\sqrt{n}\, t - x &= n \\ u_1, \ldots, u_n,\ t &\geq 0 \\ \prod_{p=1}^{n} u_p &= 0 \end{aligned}$$

hinzu. Geometrisch bedeutet dies, daß der Graph $y = t(x)$ durch einen Polygonzug angenähert wird. Die Annäherung ist um so besser, je größer n gewählt wird. Falls der Graph $y = t(x)$ eine stückweise lineare Kurve ist, erhält man sogar eine exakte Lösung. Dies sei im folgenden kurz erläutert.

2.8 Lineare Optimierungsprobleme mit zusätzlichen stückweisen linearen Funktionen:

Angenommen $y = t(x)$ sei durch die Abbildung



gegeben. Durch die Ersetzungen

$$x = u_1 + u_2 + u_3$$
$$t(x) = 5u_1 + u_2 + 3u_3$$

und Hinzufügen der Restriktionen

$$0 \leq u_1 \leq 4$$
$$0 \leq u_2 \leq 6$$
$$0 \leq u_3 \leq 5$$
$$u_2 \cdot (4-u_1) = 0$$
$$u_3 \cdot (6-u_2) = 0$$

läßt sich diese Funktion ohne Schwierigkeiten in der Form (1) - (4) darstellen.

In 2.7 und 2.8 ließe sich noch eine Vielzahl anderer Spezialprobleme angeben. Überhaupt könnte man die Aufzählung der Probleme, die sich auf die Form (1) - (4) zurückführen lassen, noch lange fortsetzen. Z.B. findet man in [Winkels, 1979] u.a. Anwendungen innerhalb der Spieltheorie. In der vorliegenden Arbeit wurden in erster Linie betriebswirtschaftliche Entscheidungstatbestände berücksichtigt, um die praktische Bedeutung der linearen Systeme mit zusätzlichen Disjunktivitätsbedingungen zu demonstrieren. Zum Abschluß dieses Paragraphen sei jedoch

noch ein Problemtyp genannt, da seine Zurückführung auf die Form (1) - (4) einige amüsante Verwirrung gestiftet hat. Es handelt sich dabei um die

2.9 Absolutwert-Optimierungsprobleme:

$$\begin{aligned} c|x| &\rightarrow \max! \\ Ax &= a \quad , \end{aligned} \tag{7}$$

dabei sei c ein beliebiges reelles N-Tupel.

Bis 1969 glaubte man, (7) mit Hilfe der gewöhnlichen Simplexmethode lösen zu können, indem man jede Variable durch die Differenz zweier vorzeichenbeschränkter Variablen ersetzte. HADLEY verlangte dafür sogar einen Beweis von seinen Lesern (siehe [Hadley, 1962, S. 172]). Die Ungültigkeit dieser Annahme[1] wurde zuerst von SHANNO und WEIL gezeigt, die gleichzeitig ihrer Verwunderung darüber Ausdruck verliehen, daß der Fehler offenbar in der gesamten Literatur unbemerkt geblieben war (siehe [Shanno/Weil, 1971]). Die Angelegenheit wurde kurz darauf von KAPLAN bagatellisiert, der eine geringfügige Modifikation der Simplexmethode zur Lösung eines Absolutwert-Optimierungsproblems angab (siehe [Kaplan, 1971]). Wenig später bewiesen RAVINDRAN und HILL die Ungültigkeit dieser Modifikation (siehe [Ravindran/Hill, 1973]).

Die Schwierigkeiten, die man übersehen hatte, resultierten daraus, daß durch die Ersetzung

$$x = u - v; \quad u,v \geq 0$$

(7) "nur" in ein System der Form (1) - (4) transformiert wird, das mit der gewöhnlichen Simplexmethode nicht gelöst werden kann. Es gilt nämlich

$$|x| = u + v$$

genau dann, wenn

$$uv = 0$$

ist.

---

1) Die Annahme ist nur dann gültig, wenn $c \leq 0$ ist.

3. Simplexverfahren

Nachdem lineare Systeme mit zusätzlichen Disjunktivitätsbedingungen dargestellt worden sind und ihre praktische Bedeutung demonstriert worden ist, stellt sich nun natürlich die Frage:

"Wie löst man solche Systeme am effizientesten?"

Für das System (1) - (3) läßt sich diese Frage nach mehr als dreißigjährigen Rechenerfahrungen eindeutig beantworten:

"Durch Anwendung des Simplexverfahrens."

Aufgrund des Satzes auf Seite 9 und der Struktur der Restriktionen (4), die die Verwendung von Basislösungen geradezu implizieren, liegt es nun nahe, das Simplexverfahren so zu erweitern, daß auch eine Anwendung auf das System (1) - (4) möglich wird. Im Hinblick hierauf faßt dieser Paragraph noch einmal kurz die Vorgehensweise bei Anwendung des Simplexverfahrens auf das System (1) - (3) zusammen. Dadurch wird die im nächsten Paragraphen vorgestellte Erweiterung für das System (1) - (4) unmittelbar verständlich.

Geht man von einem allgemeinen linearen System (1) - (3) aus und transformiert man dieses wie auf Seite 6 dargestellt auf die Gestalt (1)~ - (3)~, so kann man eine - in der Regel unzulässige - Anfangsbasislösung für das System (1)~ - (3)~ sofort durch

$$x = y - R = \begin{pmatrix} x \\ y \end{pmatrix} - R = \begin{pmatrix} x_S \\ x_L \end{pmatrix} := \begin{pmatrix} o \\ -R \end{pmatrix}$$

angeben. Die Komponenten des M-Tupels x_S heißen Strukturvariablen, die Komponenten des N-Tupels x_L logische Variablen oder Zeilenvariablen.

Entsprechend erhält man eine Anfangsbasislösung für das System (1) - (3), wenn man die Werte der Typ 2'-Variablen mit -1 multipliziert. Das weitere Vorgehen läßt sich in drei Phasen einteilen.

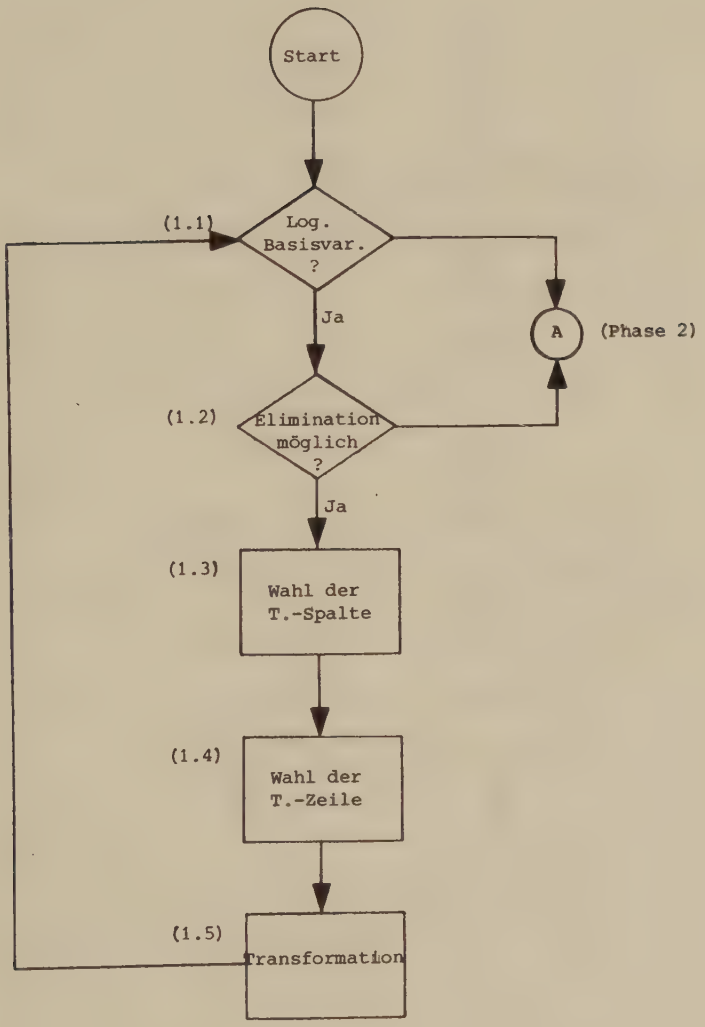
Phase 1: CRASH
In dieser Phase werden die logischen Variablen aus der Basis eliminiert.

Phase 2: DUAL
In dieser Phase wird - sofern sie existiert - eine zulässige Basislösung ermittelt, d.h. eine Basislösung von (2), die (3) erfüllt.

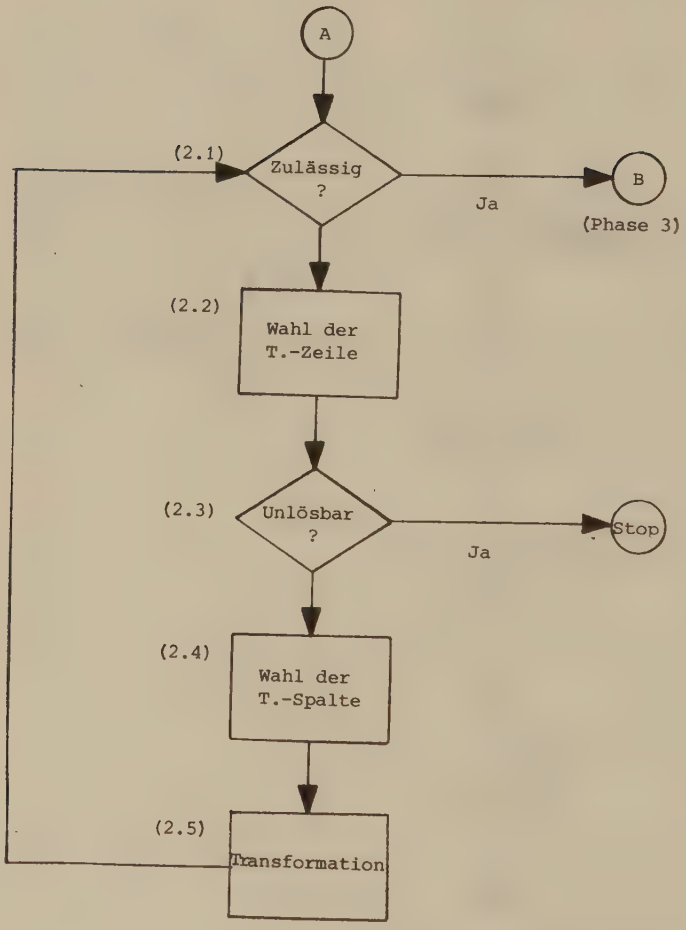
Phase 3: PRIMAL
In dieser Phase wird - sofern sie existiert - eine optimale Basislösung ermittelt, d.h. eine Basislösung von (2), die (3) erfüllt und (1) maximiert.

Im einzelnen läßt sich das Verfahren durch die folgenden Flußdiagramme veranschaulichen:

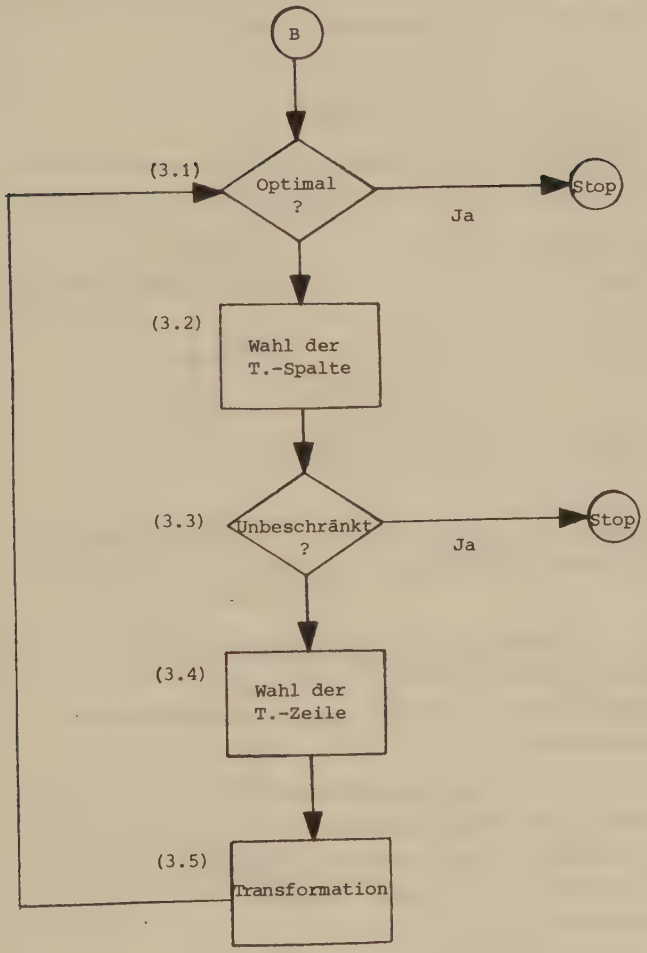
Phase 1: CRASH



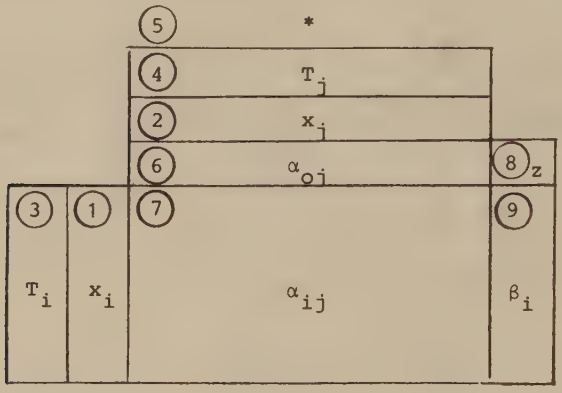
Phase 2: DUAL


A
(2.1)
Zulässig ?
Ja
B
(Phase 3)
(2.2)
Wahl der T.-Zeile
(2.3)
Unlösbar ?
Ja
Stop
(2.4)
Wahl der T.-Spalte
(2.5)
Transformation

Phase 3: PRIMAL



Die Bezeichnungen "Zeile" und "Spalte" beziehen sich dabei auf das Simplextableau, welches zur Erläuterung und Veranschaulichung der einzelnen Schritte überaus hilfreich ist, obwohl nur ein Teil der darin enthaltenen Größen bei der elektronischen Berechnung bestimmt wird.



Das zur Basislösung x mit der Basis B gehörige Simplextableau enthält in den einzelnen Blöcken (1) bis (9) folgende Daten:

(1) Die Basisvariablen x_i ($i \in I_B$: Indexfamilie der Basisvariablen).

(2) Die Nichtbasisvariablen x_j ($j \in I_N$: Indexfamilie der Nichtbasisvariablen).

(3) Die Typen T_i der Basisvariablen x_i ($i \in I_B$).

(4) Die Typen T_j der Nichtbasisvariablen x_j ($j \in I_N$).

(5) Die Markierungen * der Nichtbasisvariablen vom Typ 1, die auf ihrer oberen Grenze liegen.

(6) Die Komponenten

$$\alpha_{oj} = c_B A_B^{-1} A_j - c_j$$

des modifizierten Zieltupels ($j \in I_N$). A ist dabei die Koeffizientenmatrix und c das Zieltupel von (1) - (3).

⑦ Die Größen

$$\alpha_{ij} = {}_iA_B^{-1}A_j \qquad (i\in I_B,\ j\in I_N).$$

⑧ Den Zielwert

$$z = cx .$$

⑨ Die Werte der Basisvariablen

$$\beta_i = {}_iA_B^{-1}a - \Sigma b_j^+\alpha_{ij} \qquad (i\in I_B, j\in I_N \text{ und markiert})$$

Mit Hilfe dieser Bezeichnungen lassen sich die drei Phasen (CRASH, DUAL, PRIMAL) des Simplexverfahrens nun genauer erläutern.

Erläuterungen zu Phase 1

(1.1) Abfrage, ob sich noch logische Variablen in der Basis befinden, d.h. $I_B \cap I_L \neq \emptyset$. ($I_L$: Indexfamilie der logischen Variablen).

(1.2) Abfrage, ob eine Spalte $t\in I_N \setminus I_L$ existiert, die ein $\alpha_{rt} \neq 0$ ($r\in I_L$) enthält.

(1.3) Wahl eines solchen t's als Transformationsspalte.

(1.4) Wahl einer Zeile $r\in I_L$ als Transformationszeile, für die $\alpha_{rt} \neq 0$ gilt.

(1.5) Pivotschritt mit α_{rt} als Pivotelement, d.h. Übergang zu dem zu B' gehörigen Simplextableau, wobei B' aus B durch Austausch der Spalte A_r mit A_t hervorgeht (Basistausch).

Erläuterungen zu Phase 2

(2.1) Abfrage, ob

$$\zeta = \min(\zeta_1, \zeta_2)$$

$$\text{mit } \zeta_1 = \min_{i\in I_B \setminus I_B^3} \beta_i \ , \ \zeta_2 = \min_{i\in I_B^1} (b_i^+ - \beta_i)$$

nichtnegativ ist.
Dabei ist I_B^1 die Indexfamilie der (Typ 1)-Basisvariablen ($1\in\{1,2,3\}$).

(2.2) Wahl einer Zeile r, für die ζ angenommen wird.

(2.3) Abfrage, ob die vorliegende Basis auf die Nichtexistenz einer zulässigen Lösung hinweist, d.h. ob

$$\sigma = \begin{cases} \min(\sigma_1,\sigma_2), & \text{falls } \zeta = \zeta_1 \\ \min(\sigma_3,\sigma_4), & \text{falls } \zeta = \zeta_2 \end{cases}$$

mit

$$\sigma_1 = \min_{\substack{\alpha_{rj}<0 \\ j\in I_N \setminus I_N^{1*}}} \left|\frac{\alpha_{oj}}{\alpha_{rj}}\right|, \sigma_2 = \min_{\substack{\alpha_{rj}>0 \\ j\in I_N^{1*}\cup I_N^3}} \left|\frac{\alpha_{oj}}{\alpha_{rj}}\right|, \sigma_3 = \min_{\substack{\alpha_{rj}>0 \\ j\in I_N \setminus I_N^{1*}}} \left|\frac{\alpha_{oj}}{\alpha_{rj}}\right|, \sigma_4 = \min_{\substack{\alpha_{rj}<0 \\ j\in I_N^{1*}\cup I_N^3}} \left|\frac{\alpha_{oj}}{\alpha_{rj}}\right|$$

nicht existiert.

Dabei ist I_N^l die Indexfamilie der (Typ l)-Nichtbasisvariablen ($l\in\{1,2,3\}$) und I_N^{1*} die Indexfamilie der markierten Typ 1-Nichtbasisvariablen.

(2.4) Wahl einer Spalte t, für die σ angenommen wird.

(2.5) Transformation je nachdem welchen Wert σ hat.

(a) $\sigma = \sigma_1$

(a1) $t\notin I_N^1$: Pivotschritt mit α_{rt} als Pivotelement.

(a2) $t\in I_N^1$

(a2A) $\frac{\beta_r}{\alpha_{rt}} < b_t^+$: Siehe (a1).

(a2B) $\frac{\beta_r}{\alpha_{rt}} \geq b_t^+$: Ersetzung von β_i durch $\beta_i - \alpha_{it} b_t^+$ $(i\in I_B)$ und von z durch $z - \alpha_{ot} b_t^+$, Markierung von x_t.

(b) $\sigma = \sigma_2$

(b1) $t \in I_N^3$: Siehe (a1).

(b2) $t \in I_N^{1*}$

(b2A) $-\frac{\beta_r}{\alpha_{rt}} < b_t^+$: Löschen der Markierung von x_t, Pivotschritt mit α_{rt} als Pivotelement, Ersetzung von β_t durch $\beta_t + b_t^+$.

(b2B) $-\frac{\beta_r}{\alpha_{rt}} \geqq b_t^+$: Löschen der Markierung von x_t, Ersetzung von β_i durch $\beta_i + \alpha_{it} b_t^+ (i \in I_B)$ und von z durch $z + \alpha_{ot} b_t^+$.

(c) $\sigma = \sigma_3$

(c1) $t \notin I_N^1$: Ersetzung von β_r durch $\beta_r - b_r^+$, Pivotschritt mit α_{rt} als Pivotelement, Markierung von x_r.

(c2) $t \in I_N^1$

(c2A) $\frac{\beta_r - b_r^+}{\alpha_{rt}} < b_t^+$: Siehe (c1).

(c2B) $\frac{\beta_r - b_r^+}{\alpha_{rt}} \geqq b_t^+$: Siehe (a2B).

(d) $\sigma = \sigma_4$

(d1) $t \in I_N^3$: Siehe (c1).

(d2) $t \in I_N^{1*}$

(d2A) $-\frac{\beta_r - b_r^+}{\alpha_{rt}} < b_t^+$: Löschen der Markierung von x_t, Ersetzung von β_r durch $\beta_r - b_r^+$, Pivotschritt mit α_{rt} als Pivotelement, Ersetzung von β_t durch $\beta_t + b_t^+$, Markierung von x_r.

(d2B) $-\frac{\beta_r - b_r^+}{\alpha_{rt}} \geqq b_t^+$: Siehe (b2B).

Erläuterungen zu Phase 3

(3.1) Abfrage, ob

$$\tau = \min(\tau_1, -\tau_2)$$

mit $\tau_1 = \min_{j \in I_N \setminus I_N^{1*}} \alpha_{oj}$, $\tau_2 = \max_{j \in I_N^{1*} \cup I_N^3} \alpha_{oj}$

nichtnegativ ist.

(3.2) Wahl einer Spalte t, für die τ angenommen wird.

(3.3) Abfrage, ob die vorliegende Basis auf die Unbeschränktheit der Zielfunktion hinweist, d.h. ob

$$\delta = \begin{cases} \min(\delta_1, \delta_2, b_t^+), & \text{falls } \tau=\tau_1 \text{ und } t \in I_N^1 \\ \min(\delta_1, \delta_2), & \text{falls } \tau=\tau_1 \text{ und } t \notin I_N^1 \\ \max(\delta_3, \delta_4, -b_t^+), & \text{falls } \tau=\tau_2 \text{ und } t \in I_N^{1*} \\ \max(\delta_3, \delta_4), & \text{falls } \tau=\tau_2 \text{ und } t \notin I_N^{1*} \end{cases}$$

mit

$$\delta_1 = \min_{\substack{\alpha_{it}>0 \\ i \in I_B \setminus I_B^3}} \frac{\beta_i}{\alpha_{it}}, \quad \delta_2 = \min_{\substack{\alpha_{it}<0 \\ i \in I_B^1}} \frac{\beta_i - b_i^+}{\alpha_{it}}, \quad \delta_3 = \max_{\substack{\alpha_{it}<0 \\ i \in I_B \setminus I_B^3}} \frac{\beta_i}{\alpha_{it}}, \quad \delta_4 = \max_{\substack{\alpha_{it}>0 \\ i \in I_B^1}} \frac{\beta_i - b_i^+}{\alpha_{it}}$$

nicht existiert.

(3.4) Wahl einer Zeile r, für die δ angenommen wird. (Entfällt, falls $\delta = \pm b_t^+$.)

(3.5) Transformation je nachdem welchen Wert σ hat.

(a) $\delta = \delta_1$: Siehe (2.5), (a1).
(b) $\delta = \delta_2$: Siehe (2.5), (c1).
(c) $\delta = b_t^+$: Siehe (2.5), (a2B).
(d) $\delta = \delta_3$
  (d1) $t \in I_N^3$: Siehe (2.5), (a1).
  (d2) $t \in I_N^{1*}$: Siehe (2.5), (b2A).
(e) $\delta = \delta_4$
  (e1) $t \in I_N^3$: Siehe (2.5), (c1).
  (e2) $t \in I_N^{1*}$: Siehe (2.5), (d2A).
(f) $\delta = -b_t^+$: Siehe (2.5), (b2B).

Die soeben beschriebene Dreiphasenversion des Simplexverfahrens liegt heute fast jedem modernen LP-Computerprogramm [1] zugrunde, da sie sich numerisch am besten bewährt hat. Ihre umfangreiche Herleitung und Verifizierung findet man in der Literatur über höhere lineare Optimierung. [2] Zur Illustration sei das auf Seite 7 gegebene Problem (ohne Disjunktivitätsbedingungen) gelöst. Als Anfangslösung für die Normalform (1) - (3) (vergl. S. 8) erhält man

$$x_B = x_L = \begin{pmatrix} x_5 \\ x_6 \\ x_7 \\ x_8 \end{pmatrix} = \begin{pmatrix} 0 \\ -5 \\ 1 \\ -3 \end{pmatrix}$$

---

1) Siehe z.B. [IBM, 1979].

2) Siehe insbesondere [Orchard-Hays, 1968], teilweise auch [Neumann, 1975], [Hadley, 1962], [Collatz/Wetterling, 1966], [Zoutendijk, 1976], [Simonnard, 1966].
Der Zusammenhang mit dem "gewöhnlichen" Simplexverfahren, das auf Probleme der Form (1) - (3) beschränkt ist, in denen nur (Typ 2)-Variablen auftreten, geht auch aus dem Beweis auf Seite 61 hervor.

Dem entspricht das Simplextableau

		1	3	1	2	
		x_1	x_2	x_3	x_4	
		-1	7	-3	-5	0
1	x_5	-1	0	0	-1	0
1	x_6	1	0	-1	0	-5
1	x_7	-1	1	0	0	1
2	x_8	0	1	-1	1	-3

Elimination der logischen Variablen aus der Basis, d.h. Durchlaufen der Phase 1
(Pivotelemente in Reihenfolge: $\alpha_{54}, \alpha_{63}, \alpha_{71}, \alpha_{82}$) führt zu

		1	2	1	1	
		x_7	x_8	x_6	x_5	
		-15	8	-11	3	16
2	x_4	-1	1	-1	0	1
1	x_3	1	-1	0	-1	4
1	x_1	1	-1	1	(-1)	-1
3	x_2	2	-1	1	-1	0

In (2.1) erhält man $\zeta = \zeta_1 = \beta_1 = -1 < 0$ und somit in (2.2) $r = 1$. In (2.3) ergibt sich $\sigma = \sigma_1 = |\frac{\alpha_{05}}{\alpha_{15}}| = 3$ und somit in (2.4) $t = 5$. Gemäß (2.5), (a2A) ist daher ein Pivotschritt mit α_{15} als Pivotelement (oben eingekreist) durchzuführen. Man erhält dann:

		1	2	1	1	
		x_7	x_8	x_6	x_1	
		-12	5	-8	3	13
2	x_4	-1	1	-1	0	1
1	x_3	0	0	-1	-1	5
1	x_5	-1	1	-1	-1	1
3	x_2	1	0	0	-1	1

Wegen $\zeta = 1 > 0$ ist dieses Tableau zulässig, d.h. Phase 2 beendet.
In (3.1) ergibt sich $\tau = \tau_1 = \alpha_{07} = -12 < 0$ und somit in (3.2) $t = 7$. In (3.3) erhält man $\delta = b_7^+ = 3$ und somit gemäß (3.5), (c)

*

		1	2	1	1	
		x_7	x_8	x_6	x_1	
		-12	5	-8	3	49
2	x_4	-1	1	-1	0	4
1	x_3	0	0	-1	-1	5
1	x_5	-1	1	-1	-1	4
3	x_2	1	0	0	0	-1

Eine weitere Iteration ($t = 6$, $\delta = b_6^+$) führt zu

			*		*	
		1	2	1	1	
		x_7	x_8	x_6	x_1	
		-12	5	-8	3	65
2	x_4	-1	1	-1	0	6
1	x_3	0	0	-1	-1	7
1	x_5	-1	1	-1	-1	6
3	x_2	1	0	0	-1	-2

Wegen $\tau = 3 > 0$ ist dieses Tableau optimal.
Als zugehörige optimale Basislösung ergibt sich daraus

$$x = \begin{pmatrix} 0 \\ -2 \\ 7 \\ 6 \\ 6 \\ 2 \\ 3 \\ 0 \end{pmatrix}$$

mit dem Zielwert $z = 65$.
Für das Ausgangsproblem erhält man also die Lösung

$$\underline{x} = \begin{pmatrix} 0 \\ -2 \\ 3 \\ 6 \end{pmatrix} \quad \text{mit } \underline{z} = 53 \ .$$

4. Erweitertes Simplexverfahren

In diesem Paragraphen wird das zuvor dargestellte Simplexverfahren zur Anwendung auf das System (1) - (4) erweitert. Das daraus resultierende Vorgehen läßt sich wiederum in drei Phasen einteilen.

Phase 1:
In dieser Phase werden zunächst die logischen Variablen unter Berücksichtigung von (4) aus der Basis eliminiert.

Phase 2:
In dieser Phase wird - sofern sie existiert - eine zulässige Basislösung ermittelt, d.h. eine Basislösung von (2), die (3) und (4) erfüllt.

Phase 3:
In dieser Phase wird versucht, unter Beibehaltung der Disjunktivität eine Basislösung mit möglichst großem Zielwert zu bestimmen. Ist ein Übergang zu einer Basislösung mit größerem Zielwert unter Beibehaltung der Disjunktivität nicht mehr möglich, so wird die Phase 2 erneut durchlaufen. Zuvor wird jedoch ein Mindestzielwert festgelegt, der den besten bisher gefundenen Zielwert um einen positiven Betrag übersteigt (JUMP, siehe 4.1.3).

4.1 Grundbestandteile des Verfahrens

In diesem Abschnitt werden die auf der Simplexmethode aufbauenden Grundbestandteile des vorzustellenden Verfahrens angegeben. Die Art und Weise ihrer Kombination und Integration ist Gegenstand des nächsten Abschnittes.

4.1.1 Zusatzregeln

Würde man die Simplexmethode unmodifiziert auf das System (1) - (4) anwenden, so wären in Kürze die Disjunktivitätsbedingungen verletzt, da sie bei den Transformationsschritten nicht miteinbezogen werden. Im folgenden sollen deshalb Regeln formuliert werden, durch die (4) bei der Auswahl der

Transformationsspalte und Zeile sowie bei der Transformation selber mitberücksichtigt wird. Dazu seien zunächst einige neue Begriffe eingeführt.

Definition: Sei $t \in U\{N_k^- \cup N_k^+ : k \in K\}$.
Für eine disjunktive Restriktion $k_o \in K$ heiße O ein relevanter Rand von x_t, falls $t \in N_{k_o}^-$ ist, und b_t^+ ein relevanter Rand, falls $t \in N_{k_o}^+$ ist. Bezüglich einer Basislösung x mit der Basis B heiße t kritisch, wenn es ein $k_o \in K$ gibt, für das x_t die einzige auf einem relevanten Rand liegende Nichtbasisvariable ist. B heiße kritisch, wenn es ein $k_o \in K$ gibt, für das es keine auf einem relevanten Rand liegende Nichtbasisvariable gibt.

Der Sinn dieser Definition wird aus den folgenden Regeln ersichtlich, die nun in Abhängigkeit von bestimmten vorliegenden Basislösungen Anwendung finden:

Regel 1: Benutze nur unkritische Transformationsspalten.

Die Verwendung der Regel 1 garantiert, daß im Verlaufe des Simplexverfahrens die Restriktionen (4) stets erfüllt bleiben, wenn man von einer disjunktiven Basislösung (d.h. einer Basislösung von (2), die (4), aber nicht notwendigerweise (3) erfüllt) mit unkritischer Basis ausgeht.

Zur Illustration sei das Problem von Seite 8 betrachtet. Das Anfangstableau war (vergl. S. 32)

		1	3	1	2	
		x_1	x_2	x_3	x_4	
		-1	7	-3	-5	0
1	x_5	-1	0	0	-1	0
1	x_6	1	0	-1	0	-5
1	x_7	-1	1	0	0	1
2	x_8	0	1	-1	1	-3

Die zugehörige Basislösung ist disjunktiv, da die Restriktionen

$$
\begin{aligned}
(7-x_5) \quad \cdot x_4 &= 0 \\
(2-x_1) \quad \cdot x_3 &= 0 \\
(8-x_3) \cdot x_1 \cdot x_3 &= 0 \\
(3-x_7) \cdot x_4 \cdot x_6 &= 0
\end{aligned}
$$

erfüllt sind. Ferner ist die Basis unkritisch, da für alle $k \in K = (1,2,3,4)$ mindestens eine Nichtbasisvariable auf einem relevanten Rand liegt. Bei der Elimination der logischen Variablen aus der Basis ist nun zusätzlich die Regel 1 zu beachten. Beispielsweise erhält man nach Pivotschritten mit α_{51} und α_{72} als Pivotelemente das Tableau

		1	1	1	2	
		x_5	x_7	x_3	x_4	
		6	-7	-3	-11	-7
1	x_1	-1	0	0	1	0
1	x_6	1	0	-1	-1	-5
3	x_2	-1	1	0	1	1
2	x_8	1	-1	-1	0	-4

Da in diesem Tableau die Spalten 3 und 4 kritisch sind, ist eine weitere Elimination der logischen Variablen unter Berücksichtigung der Regel 1 aus der Basis nicht mehr möglich und die Phase 1 damit beendet. In Phase 2 ergäbe sich r = 6 und t = 3. Da t = 3 jedoch kritisch ist, ist in (2.1) für ζ der zweitkleinste negative Wert zu wählen usw., bis man eine Transformationszeile gefunden hat, deren gemäß (2.4) bestimmte Transformationsspalte unkritisch ist. Ist dies nicht möglich, läßt sich Regel 1 wie folgt abschwächen:

Regel 2: Benutze kritische Transformationsspalten nur dann, wenn keine geeigneten unkritischen existieren.

Regel 3: Berücksichtige bei der Minimumsbildung zur Auswahl der Transformationsspalte innerhalb eines dualen Simplexschrittes nur unkritische Spalten.

Regel 3 besagt, daß σ zusätzlich nur über unkritische Spalten bestimmt wird. Im obigen Tableau hat man also für r = 6 kein geeignetes t, jedoch für r = 8 ergibt sich t = 7. Gemäß (2.5), (a2B) erhält man dann

			*			
		1	1	1	2	
		x_5	x_7	x_3	x_4	
		6	-7	-3	-11	14
1	x_1	-1	0	0	1	0
1	x_6	1	0	(-1)	-1	-5
3	x_2	-1	1	0	1	-2
2	x_8	1	-1	-1	0	-1

Nun läßt sich kein dualer Simplexschritt unter Berücksichtigung der Regel 1 bzw. Regel 3 mehr durchführen. Ein gewöhnlicher dualer Simplexschritt (Pivotelement eingekreist) ergibt

			*			
		1	1	1	2	
		x_5	x_7	x_6	x_4	
		3	-7	-3	-8	29
1	x_1	-1	0	0	1	0
1	x_3	-1	0	-1	1	5
3	x_2	-1	1	0	1	-2
2	x_8	0	-1	-1	1	4

Die zugehörige Basislösung erfüllt (3), ist jedoch nicht mehr disjunktiv, da (4) für k = 2 verletzt ist. Weiterhin ist die Basis für k = 2 und k = 3 kritisch. Zur Wiederherstellung der Disjunktivität und einer unkritischen Basis dienen die folgenden Regeln:

Regel 4: Ist die Basis für eine disjunktive Restriktion kritisch und liegt eine in dieser disjunktiven Restriktion enthaltene Basisvariable auf einem relevanten Rand, so eliminiere diese Variable aus der Basis unter Verwendung eines Zufallsgenerators und unter Beachtung von Regel 1.

Regel 5: Ist die Basis für eine disjunktive Restriktion kritisch, so setze eine in dieser disjunktiven Restriktion enthaltene Nichtbasisvariable auf einen relevanten Rand unter Verwendung eines Zufallsgenerators.

Regel 6: Ist die Basis für eine disjunktive Restriktion kritisch, so eliminiere eine in dieser disjunktiven Restriktion enthaltene Basisvariable aus der Basis unter Verwendung eines Zufallsgenerators und unter Beachtung von Regel 1 und setze sie auf einen relevanten Rand.

Die Anwendung der Regel 4 auf das obige Tableau ergibt also

			*			
		1	1	1	2	
		x_1	x_7	x_6	x_4	
		3	-7	-3	-5	29
1	x_5	-1	0	0	-1	0
1	x_3	-1	0	-1	0	5
3	x_2	-1	1	0	0	-2
2	x_8	0	-1	-1	1	4

Da hierbei t = 5 die einzige unkritische Spalte mit nichtverschwindendem Pivotelement war, kam nur sie als Transformationsspalte in Frage. Andernfalls wäre die Auswahl zufällig erfolgt. Dadurch soll ein Kreisen, d.h. eine sich dauernd wiederholende Iterationsfolge vermieden werden. Da die Regeln 4 - 6 nur den Zweck haben, die Disjunktivität und eine unkritische Basis wiederherzustellen, geschieht die Auswahl des Privotelementes (entfällt bei Regel 5) wie in Phase 1 ohne jeden Rechenaufwand, so daß bei der elektronischen Berechnung eine hohe Iterationsanzahl innerhalb kürzester Zeit möglich wird.

4.1.2 REDUCE

Betrachtet man das letzte Simplextableau, dann fällt auf, daß die Zeile 3 das Minimumskriterium $\tau \leq 0$ erfüllt, wenn man sie als modifizierte Zielzeile zur Zielfunktion $x_3 \rightarrow \min!$ auffaßt. Daraus folgt, [1] daß in jeder zulässigen Lösung des auf Seite 8 gegebenen Problems (1) - (3) automatisch $x_3 \geq 5$ ist. Diese Tatsache kann man sich zur Reduktion sowohl der Arbeitsmatrix

1) Siehe [Telgen, 1979], [Thompson/Tonge/Zionts, 1966].

als auch der disjunktiven Restriktionen (4) zu Nutze machen. Für eine Variable x_j vom Typ 1 oder 2 ist dabei die Randbedingung $0 \leqq x_j$ streng redundant, wenn im obigen Sinne $\min x_j > 0$ ist, und schlupflos, wenn $\max x_j = 0$ ist. Entsprechend ist für eine Variable vom Typ 1 die Randbedingung $x_j \leqq b_j^+$ streng redundant, wenn $\max x_j < b_j^+$ ist, und schlupflos, wenn $\min x_j = b_j^+$ ist.

Streng redundante und schlupflose Randbedingungen lassen sich demnach bei linear zulässigen Basislösungen (d.h. Basislösungen, die (3), aber nicht notwendigerweise (4) erfüllen) wie oben beschrieben, durch das Abtesten der jeweiligen Optimalitätskriterien ermitteln. Man kann nun streng redundante Randbedingungen durch Freigabe der entsprechenden Variablen (Typänderung) oder durch Streichen der entsprechenden Zeilen im Simplextableau (falls $j \in I_L$ und $\min x_j > 0$) eliminieren und die zugehörigen Faktoren in (4) auslassen. Sollte sich dabei eine disjunktive Restriktion auf einen einzigen Faktor reduzieren lassen, so wird die zugehörige Variable auf den entsprechenden relevanten Rand fixiert und die disjunktive Restriktion gestrichen. Ergibt sich dabei ein Widerspruch - beispielsweise $\min x_j > 0$ nach Fixierung von x_j auf Null -, so ist das Problem (1) - (4) unlösbar und das Verfahren beendet. Beim Auftreten von schlupflosen Randbedingungen fixiert man die zugehörige Variable auf diesen Rand und eliminiert alle disjunktiven Restriktionen, die den entsprechenden Faktor aufweisen.

Aufgrund der obigen Ausführungen können im Zahlenbeispiel die disjunktiven Restriktionen 2 und 3 gestrichen und die Variablen x_1 und x_3 auf ihre oberen Grenzen fixiert werden. Man erhält dann nach Umrechnung des letzten Tableaus gemäß (3.5), (c) und anschließend (3.5), (b)

⊡ * ⊡

		1	1	1	2	
		x_1	x_7	x_3	x_4	
		6	-7	-3	-5	26
1	x_5	-1	0	0	-1	2
1	x_6	1	0	-1	0	1
3	x_2	-1	1	0	0	0
2	x_8	1	-1	-1	1	5

Die auf ihren oberen Grenzen fixierten Variablen werden mit ⊡ markiert, die auf Null fixierten mit □.

4.1.3 JUMP

Dem soeben erhaltenen Tableau entspricht eine linear zulässige und disjunktive, d.h. zulässige, Basislösung des auf Seite 8 gegebenen Problems (1) - (4). Man kann also die Phase 3 beginnen.

Da ein Übergang zu einer Basislösung mit größerem Zielwert unter Beibehaltung der Disjunktivität, d.h. ein primaler Simplexschritt unter Beachtung der Regel 1, nicht mehr möglich ist, speichert man die zuletzt gefundene zulässige Lösung $\hat{x}$ und ihren Zielwert $\hat{z}$. Daraufhin beginnt man die Phase 2 aufs neue und sucht nach einer zulässigen Lösung von (1) - (4), für deren Zielwert z gelten muß:

$$\hat{z} + \hat{\lambda} \leq z. \qquad (8)$$

Dabei ist $\hat{\lambda} > 0$ eine vom Benutzer festzulegende Schrittweite, die natürlich in Abhängigkeit von $\hat{z}$ variiert werden kann. Gibt es keine solche zulässige Lösung mehr, so gilt für den optimalen Zielwert z^* von (1) - (4) die Abschätzung

$$\hat{z} \leq z^* \leq \hat{z} + \hat{\lambda}. \qquad (9)$$

Bei der numerischen Durchführung dieses "Sprunges auf ein erhöhtes Anspruchsniveau" fügt man dem letzten Simplextableau die modifizierte Zielzeile mit der (Typ 2)-Zeilenvariablen x_o und $\beta_o = -\hat{\lambda}$ als zusätzliche Restriktionszeile hinzu. Dem entspricht in der Normalform (1) - (4) die zusätzliche Restriktion

$$cx - x_o = \hat{z} + \hat{\lambda},$$

also, wegen $x_o \geqq 0$

$$z = cx \geqq \hat{z} + \hat{\lambda},$$

so daß (8) gewährleistet ist.

Im Zahlenbeispiel ergibt sich demnach mit $\hat{\lambda} = 10$

		⊡	*	⊡		
		1	1	1	2	
		x_1	x_7	x_3	x_4	
		6	-7	-3	-5	26
1	x_9	-1	0	0	-1	2
1	x_6	1	0	-1	0	1
3	x_2	-1	1	0	0	0
2	x_8	1	-1	-1	1	5
2	x_o	6	-7	-3	(-5)	-10

Ein dualer Simplexschritt (Pivotelement ist eingekreist) führt zu

		⊡	*	⊡		
		1	1	1	2	
		x_1	x_7	x_3	x_o	
		0	0	0	-1	36
1	x_5	-2,2	1,4	0,6	-0,2	4
1	x_6	1	0	-1	0	1
3	x_2	-1	1	0	0	0
2	x_8	2,2	-2,4	-1,6	0,2	3
2	x_4	-1,2	1,4	0,6	-0,2	2

und ein weiterer Transformationsschritt nach Regel 6 zu

		⊡	*	⊡	*	
		1	1	1	1	
		x_1	x_7	x_3	x_5	
		11	-7	-3	-5	51
2	x_o	11	-7	-3	-5	15
1	x_6	1	0	-1	0	1
3	x_2	-1	1	0	0	0
2	x_8	0	-1	1	1	0
2	x_4	1	0	0	-1	5

Da die entsprechende Basislösung zulässig ist, ist die Phase 2 wiederum beendet. Die nächste Ausführung von JUMP erfordert nur noch die Ersetzung von β_o durch $-\lambda$.

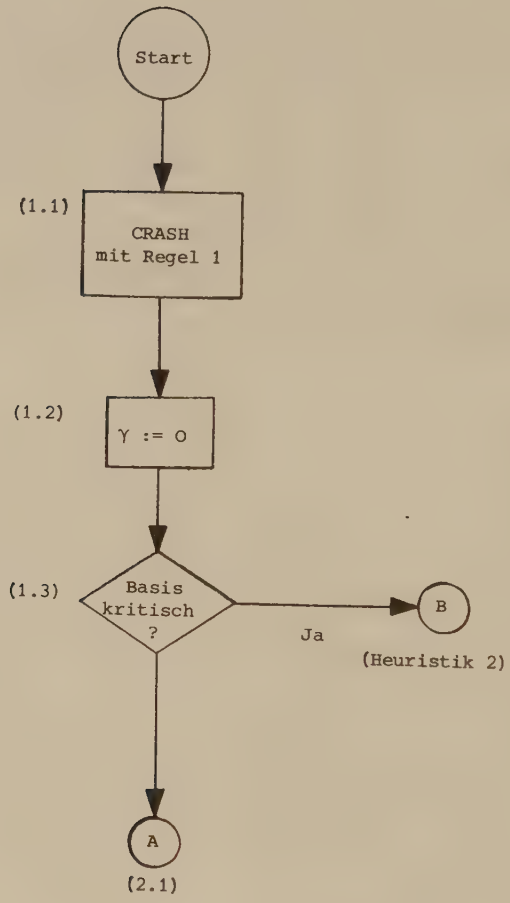
4.2 Verfahren

Wie bereits in den vorangegangenen Abschnitten angedeutet wurde, lassen sich die Zusatzregeln sowie die Prozeduren REDUCE und JUMP mit den in Paragraph 3 dargestellten Algorithmen CRASH, DUAL und PRIMAL so verbinden, daß eine Anwendung auf das Problem (1) - (4) möglich wird. CRASH, DUAL und PRIMAL werden dabei so-

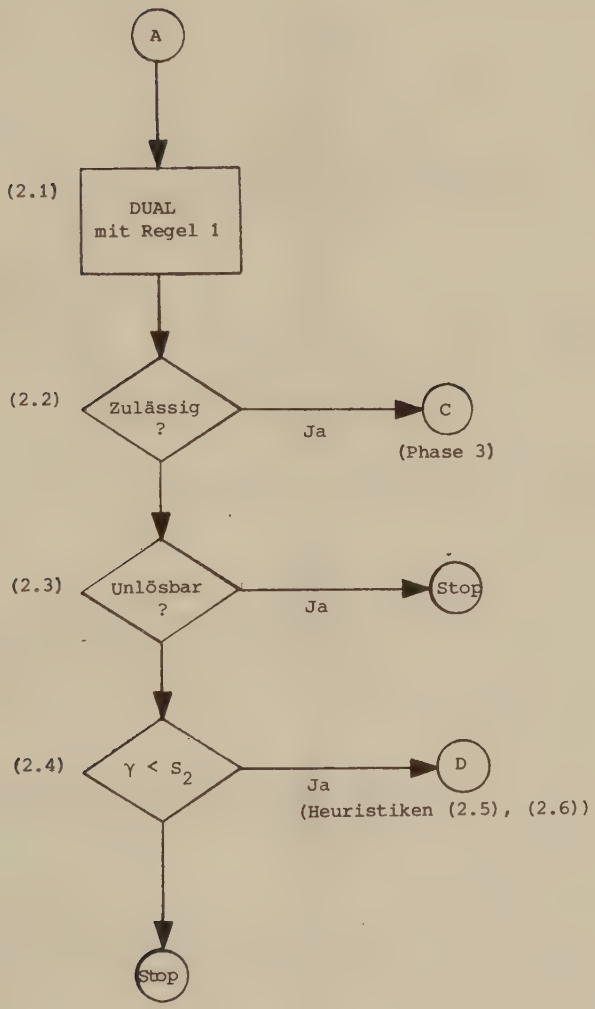
lange wie möglich unter zusätzlicher Beachtung der Regel 1 durchgeführt. In Phase 2 wird danach die Regel 1 durch Regel 3 abgeschwächt (Heuristik 1, siehe S. 48). Bleibt diese Strategie erfolglos, werden auch kritische Transformationsspalten zugelassen und anschließend wird - falls nötig - die Disjunktivität mit Hilfe von Regel 4 - 6 wiederhergestellt (Heuristik 2, siehe S. 49). In Phase 3 wird, falls keine Transformation unter Beachtung der Regel 1 mehr möglich ist, mit JUMP erneut in die Phase 2 gesprungen. Auf Wunsch kann zusätzlich die Prozedur REDUCE angewendet werden.

Im einzelnen läßt sich das Verfahren durch die folgenden Flußdiagramme veranschaulichen.

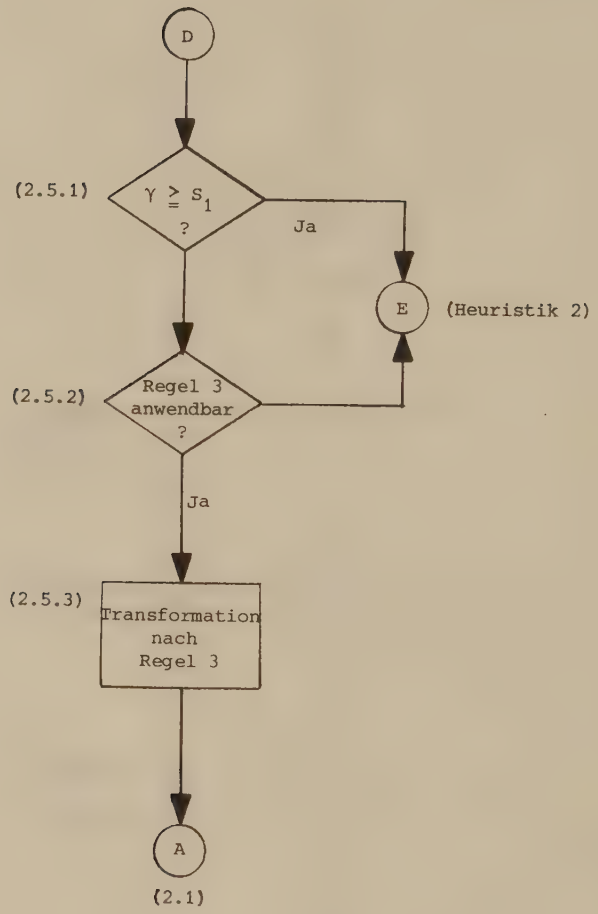
Phase 1


Start
(1.1)
CRASH
mit Regel 1
(1.2)
γ := 0
(1.3)
Basis
kritisch
?
Ja
B
(Heuristik 2)
A
(2.1)

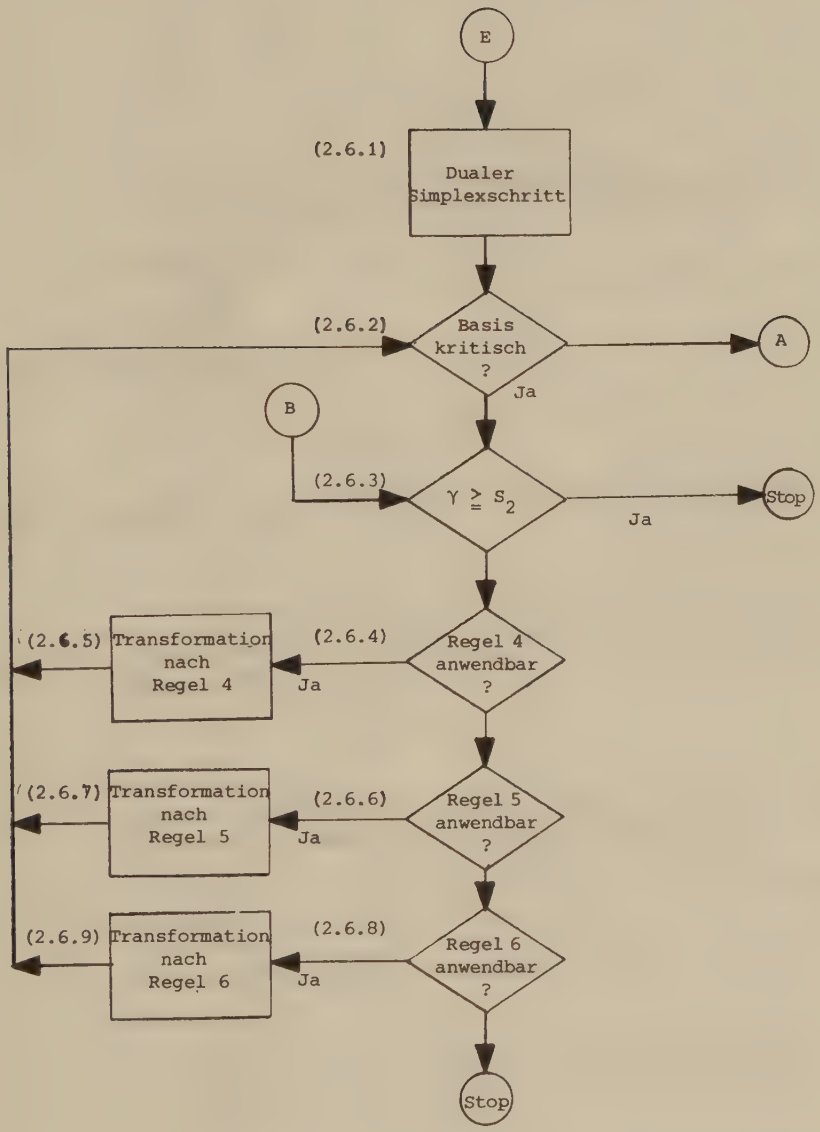
Phase 2


A
(2.1)
DUAL
mit Regel 1
(2.2)
Zulässig
?
Ja
C
(Phase 3)
(2.3)
Unlösbar
?
Ja
Stop
(2.4)
$\gamma < s_2$
Ja
D
(Heuristiken (2.5), (2.6))
Stop

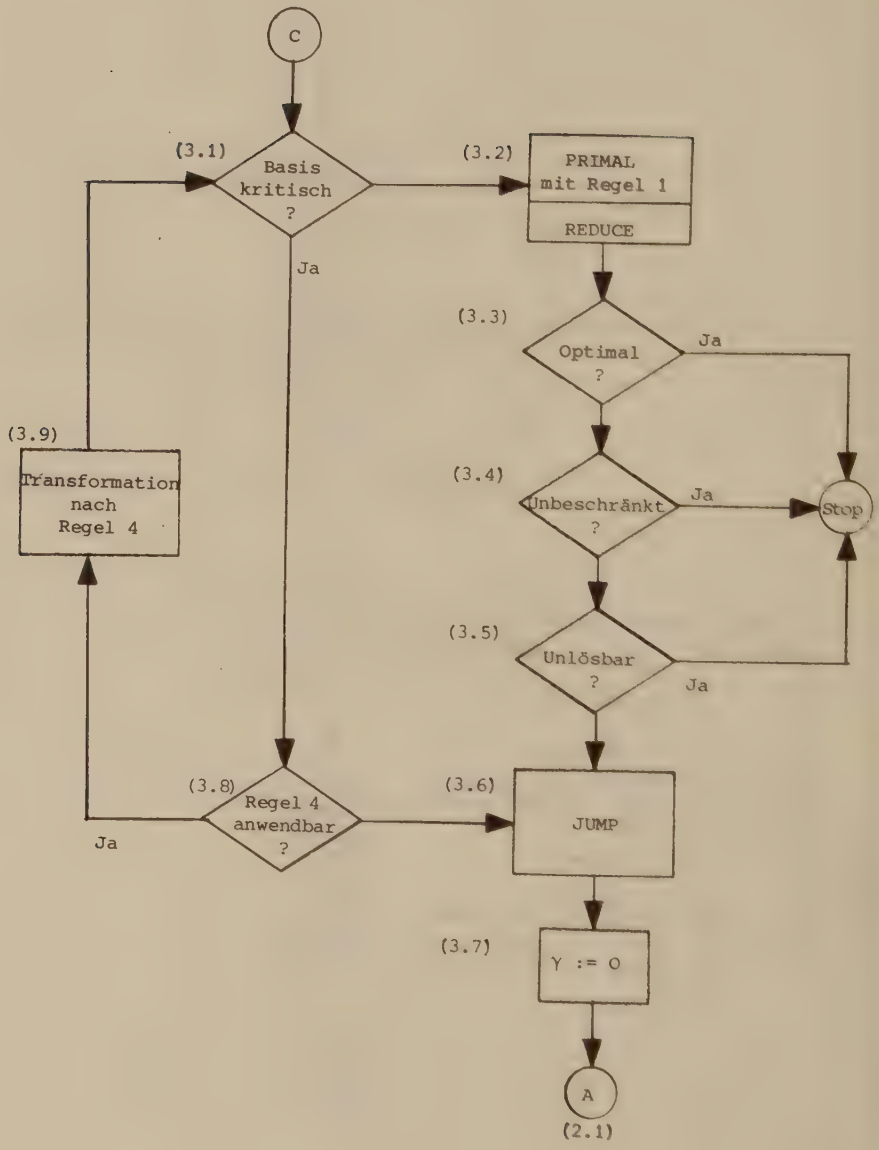
Heuristik 1


D
(2.5.1)
$\gamma \geq s_1$
?
Ja
E
(Heuristik 2)
(2.5.2)
Regel 3
anwendbar
?
Ja
(2.5.3)
Transformation
nach
Regel 3
A
(2.1)

Heuristik 2


E
(2.6.1)
Dualer Simplexschritt
(2.6.2)
Basis kritisch ?
Ja
A
B
(2.6.3)
$\gamma \geq S_2$
Ja
Stop
(2.6.4)
Regel 4 anwendbar ?
Ja
(2.6.5)
Transformation nach Regel 4
(2.6.6)
Regel 5 anwendbar ?
Ja
(2.6.7)
Transformation nach Regel 5
(2.6.8)
Regel 6 anwendbar ?
Ja
(2.6.9)
Transformation nach Regel 6
Stop

Phase 3


C
(3.1)
Basis kritisch ?
Ja
(3.2)
PRIMAL mit Regel 1
REDUCE
(3.3)
Optimal ?
Ja
(3.4)
Unbeschränkt ?
Ja
Stop
(3.5)
Unlösbar ?
Ja
(3.6)
JUMP
(3.7)
Y := 0
A
(2.1)
(3.8)
Regel 4 anwendbar ?
Ja
(3.9)
Transformation nach Regel 4

Erläuterungen zu Phase 1

(1.1) Elimination der logischen Variablen aus der Ausgangsbasis (CRASH, siehe S. 23) unter Beachtung der Regel 1 (siehe S. 36).

(1.2) Beginn der Zählung der vom Benutzer verwendeten Steuergrößen (z.B. Zeit, Anzahl der Iterationen, Anzahl der Abbrüche in Phase 2 etc.), mit deren Hilfe festgelegt werden kann, wie lange welche heuristische Strategie angewandt werden soll.

(1.3) Abfrage, ob die durch (1.1) gelieferte Basis kritisch ist.

Erläuterungen zu Phase 2

(2.1) Anwendung von DUAL (siehe S. 24) unter Beachtung der Regel 1 bis zum Abbruch.

(2.2) Abfrage, ob die vorliegende Lösung (3) und (4) erfüllt (vergl. (2.1) auf S. 27).

(2.3) Abfrage, ob die vorliegende Basis auf die Nichtexistenz einer zulässigen Lösung hinweist (vergl. (2.3) auf S. 28).

(2.4) Abfrage nach Überschreitung der Zeit- oder Iterationsschranke etc. (siehe (1.2)) für das Durchlaufen der heuristischen Strategien.

(2.5) Heuristik 1.

(2.5.1) Abfrage nach Überschreitung der Zeit- oder Iterationsschranke etc. für das Durchlaufen der Heuristik 1.

(2.5.2) Abfrage, ob ein dualer Simplexschritt unter Beachtung der Regel 3 (siehe S. 38) möglich ist.

(2.5.3) Dualer Simplexschritt unter Beachtung der Regel 3.

(2.6) Heuristik 2.

(2.6.1) Durchführung eines gewöhnlichen Iterationsschrittes (d.h. ohne Beachtung der Regel 1) in DUAL.

(2.6.2) Abfrage, ob die vorliegende Basis kritisch ist.

(2.6.3) Abfrage nach Überschreitung der Zeit- oder Iterationsschranke etc. für das Durchlaufen der Heuristik 2 ($S_1 \leq S_2$).

(2.6.4) Abfrage, ob ein Transformationsschritt nach Regel 4 (siehe S. 39) möglich ist.

(2.6.5) Transformationsschritt nach Regel 4.

(2.6.6) Abfrage, ob ein Transformationsschritt nach Regel 5 (siehe S. 39) möglich ist.

(2.6.7) Transformationsschritt nach Regel 5.

(2.6.8) Abfrage, ob ein Transformationsschritt nach Regel 6 (siehe S. 39) möglich ist.

(2.6.9) Transformationsschritt nach Regel 6.

Erläuterungen zu Phase 3

(3.1) Abfrage, ob die vorliegende Basis kritisch ist. Diese Abfrage ist zu Beginn der Phase 3 nötig, da durchaus zulässige Basislösungen von (1) - (4) mit kritischer Basis auftreten können, die Regel 1 die Erhaltung der Disjunktivität jedoch nur bei unkritischer Basis garantiert. Falls (3.1) zutrifft, muß deshalb zunächst mit Hilfe der Regel 4 eine unkritische Basis hergestellt werden.

(3.2) Anwendung von PRIMAL (siehe S. 25) unter Beachtung der Regel 1 bis zum Abbruch. Auf Wunsch kann an dieser Stelle zusätzlich die Prozedur REDUCE (siehe Abschnitt 4.1.3) benutzt werden.

(3.3) Abfrage, ob die vorliegende Lösung optimal ist (vergl. (3.1) auf S. 30). Natürlich ist eine optimale Lösung von (1) - (3) auch optimale Lösung von (1) - (4)).

(3.4) Abfrage, ob die vorliegende Basis auf die Unbeschränktheit der Zielfunktion hinweist (vergl. (3.3) auf S. 30).

(3.5) Abfrage, ob sich bei Anwendung der Prozedur REDUCE ein Widerspruch (siehe S. 41) ergeben hat. (Entfällt, falls REDUCE nicht benutzt wird.)

(3.6) Anwendung der Prozedur JUMP (siehe Abschnitt 4.1.2).

(3.7) Neubeginn der Zählung der Steuergrößen.

(3.8) Abfrage, ob ein Transformationsschritt nach Regel 4 möglich ist.

(3.9) Transformationsschritt nach Regel 4.

Eine schematische Anwendung des soeben beschriebenen Verfahrens auf das Problem von S. 8 führt zu der gleichen Tableaufolge wie in 4.1.

5. Aus der ganzzahligen Optimierung abgeleitetes allgemeines Lösungsverfahren

Mit Hilfe des im vorangegangenen Paragraphen beschriebenen Verfahrens lassen sich auch sehr große Systeme der Form (1) - (4) hinsichtlich der Rechenzeit effizient bearbeiten, da es eine einfache Erweiterung der bewährten Simplexmethode ist. Offen bleibt jedoch die Frage nach der Qualität der so gefundenen Lösung. Die Abschätzung (9) auf Seite 42 gilt nämlich nur dann, wenn bewiesen ist, daß es keine zulässige Lösung von (1) - (4) gibt, deren Zielwert größer als $\hat{z} + \hat{\lambda}$ ist. Das Nichtauffinden einer solchen Lösung in Phase 2 innerhalb der vorgegebenen Zeitschranke, Iterationsschranke etc. und damit der Abbruch des Verfahrens ist aber keinesfalls gleichbedeutend mit der Nichtexistenz einer solchen Lösung, wenn auch die Wahrscheinlichkeit dafür mit S_2 wächst. Der Benutzer muß jedoch die Möglichkeit haben, die Lösungsqualität zu beurteilen, d.h. die ebengenannte Schwäche des Verfahrens muß beseitigt werden. Aus diesem Grunde soll zunächst geprüft werden, welche weiteren vielversprechenden Ansätze sich zur Lösung von Systemen der Form (1) - (4) anbieten. Hierzu sei an die Anwendungsbeispiele 2.1 (siehe S. 11) und 2.4 - 2.6 (siehe S. 14 ff.) erinnert, die nahelegen, Lösungsprinzipien aus der ganzzahligen Optimierung zu verwenden, da diese Beispiele die enge Beziehung zwischen linearen Systemen mit zusätzlichen Disjunktivitätsbedingungen und gemischt-ganzzahligen O/1-Optimierungssystemen aufzeigen.

In dem nun folgenden Paragraphen werden daher die algorithmischen Grundprinzipien der ganzzahligen Optimierung dargestellt, und daraus wird ein allgemeines Verfahren zur Lösung des Systems (1) - (4) abgeleitet. Spezifizierungen erfolgen in den nächsten Paragraphen.

5.1 Grundbestandteile des Verfahrens

Es werden die folgenden Bezeichnungen verwendet:

(P) : Problem (1) - (4),
F(P) : Menge der zulässigen Lösungen von (P),
O(P) : Menge der optimalen Lösungen von (P),
$z^*(P)$: Optimaler Zielwert von (P),
$\hat{z}$: Bisher (im Verlauf des Algorithmus) bester Zielwert,
C : Menge aller möglichen Kandidatenprobleme (wird noch erläutert) für den Algorithmus,
$(\pi) \in C$: Ausgewähltes Kandidatenproblem.

Man benutzt nun die folgenden algorithmischen Prinzipien:

5.1.1 Relaxation

(P_R) heißt relaxiertes Problem von (P) genau dann, wenn $F(P) \subseteq F(P_R)$ gilt. Daraus folgt unmittelbar:

(a1) $F(P_R) = \emptyset \Rightarrow F(P) = \emptyset$,
(a2) $z^*(P_R) \geq z^*(P)$,
(a3) $x \in O(P_R)$ und $x \in F(P) \Rightarrow x \in O(P)$.

Das Prinzip der Relaxation wird in erster Linie bei Schnittebenenverfahren (Cutting-Plane-Verfahren) angewendet.

5.1.2 Separation

(P) wird separiert in Teilsysteme $(P_1), \ldots, (P_l)$, so daß gilt:

(b1) $\bigcup_{i=1}^{l} (P_i) = (P)$.

Diese mengentheoretische Betrachtungsweise von (P) wird kanonisch über die zulässige Menge F(P) definiert
(z.B. $(P_i) \subseteq (P) :\Longleftrightarrow F(P_i) \subseteq F(P)$). Die oben definierte Menge C ist dann eine Teilmenge der Potenzmenge $\mathcal{P}(P)$.

Das Prinzip der Separation wird in erster Linie bei Branch-and-Bound-Verfahren angewendet.

5.1.3 Verwerfen

Das Prinzip des Verwerfens hat drei mögliche Kriterien, die sich unmittelbar aus den Eigenschaften (a1), (a2) und (a3) der Relaxation ergeben. Sei $(\pi) \in C$ ein beliebiges Kandidatenproblem und es läge einer der drei folgenden Fälle vor:

(c1) Analyse von (π_R) ergibt $F(\pi) = \emptyset$,

(c2) Analyse von (π_R) ergibt $z^*(\pi) \leqq \hat{z}$,

(c3) Analyse von (π_R) ergibt $x \in O(\pi_R) \cap F(\pi)$.

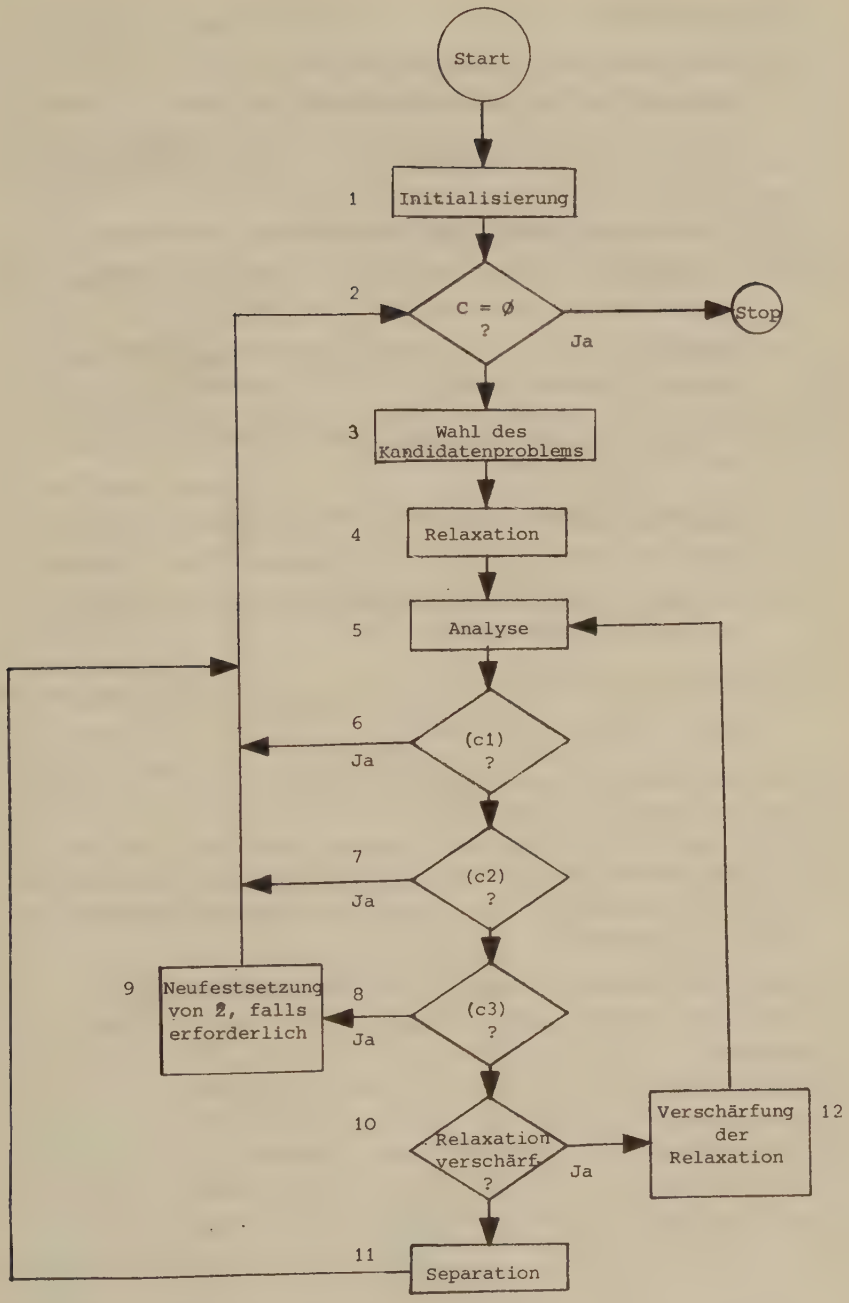
In allen drei Fällen kann (π) verworfen werden, wobei bei (c3) das Problem (π) gelöst ist und dessen Lösung $x \in O(\pi)$ mögliche Lösung von (P) sein kann.

5.2 Verfahren

Unter Benutzung der obengenannten drei Prinzipien läßt sich dann ein allgemeiner Algorithmus für das Problem (1) - (4) durch das folgende Flußdiagramm veranschaulichen: [1)]

---

1) Vergl. [Geoffrion/Marsten, 1972].


Start
1
Initialisierung
2
C = ∅ ?
Ja
Stop
3
Wahl des Kandidatenproblems
4
Relaxation
5
Analyse
6
(c1) ?
Ja
7
(c2) ?
Ja
8
(c3) ?
Ja
9
Neufestsetzung von 2, falls erforderlich
10
Relaxation verschärf. ?
Ja
12
Verschärfung der Relaxation
11
Separation

Erläuterungen

1. Initialisierung: $C := \{(P)\}$; $\hat{z} := -\infty$
2. Abfrage, ob die Kandidatenliste leer ist.
3. Wahl eines Kandidatenproblems $(\pi) \in C$ und Neufestsetzung von $C := C \setminus \{(\pi)\}$.
4. Bildung des relaxierten Problems (π_R) von (π).
5. Lösung von (π_R) bzw. Verwendung eines Algorithmus mit hinreichender Bedingung für (c1), (c2), (c3).
6. Abfrage, ob der erste Fall des Verwerfens vorliegt, d.h. ob $F(\pi_R) = \emptyset$ ist.
7. Abfrage, ob der zweite Fall des Verwerfens vorliegt, d.h. ob $z^*(\pi_R) \leq \hat{z}$ ist.
8. Abfrage, ob der dritte Fall des Verwerfens vorliegt.
9. Neufestsetzung von $\hat{z} := z^*(\pi)$, falls $z^*(\pi) > \hat{z}$ gilt.
10. Abfrage, ob eine Verschärfung der Relaxation sinnvoll ist.
11. Separation von (π) in $(\pi_{j_1}), \ldots, (\pi_{j_L})$ und Neufestsetzung von $C := C \cup \{(\pi_{j_1}), \ldots, (\pi_{j_L})\}$.
12. Verschärfung der Relaxation, d.h. Bildung von $(\pi_{R'})$ mit $F(\pi) \subset F(\pi_{R'}) \subset F(\pi_R)$.

Für das soeben beschriebene Verfahren zur Lösung von (1) - (4) folgt unmittelbar aus den Definitionen von Relaxation, Separation und Verwerfen:

Satz: Das Verfahren endet mit einer optimalen Lösung von (1) - (4) oder mit der Aussage, daß (1) - (4) keine zulässige Lösung besitzt.

6. Schnittebenenverfahren

Entsprechend der aus der ganzzahligen Optimierung bekannten Definition ist eine Schnittebene (engl.: Cutting Plane), in dem hier vorliegenden Fall grob vereinfachend gesprochen eine zusätzliche Restriktion, die mindestens eine Basislösung von dem durch (2) und (3) gebildeten (eventuell bereits durch andere Schnittebenen reduzierten) Bereich "abschneidet" ohne dabei eine disjunktive Lösung "mitabzuschneiden" (zulässiger Schnitt, vergl. auch S. 60).

Die zahlreichen Modifikationen des Schnittebenenverfahrens bestehen in der Konstruktion der Schnitte und der daraus abgeleiteten Konvergenz des Verfahrens. Das allgemeine Algorithmusschema auf Seite 57 muß für das Schnittebenenverfahren wie folgt spezifiziert werden:

3. Festsetzung von $(\Pi) := (P)$. (Schritt wird nur einmal durchlaufen.)
4. Relaxation der Disjunktivitätsbedingungen.
5. Lösung von (Π_R) mit Hilfe des Simplexverfahrens.
7. Entfällt.
10. Übergang stets zu Schritt 12.
11. Entfällt.
12. Hinzufügen einer Schnittebene.

Diejenigen Schritte, die oben nicht im einzelnen aufgeführt sind, ergeben sich kanonisch aus dem allgemeinen Verfahren von Seite 57. Schritt 12 besagt, daß dem in Schritt 5 berechneten optimalen Simplextableau die Zeile

2	x_q	α_{qj}	β_q

hinzugefügt wird, wobei $q = |I_B| + |I_N| + 1$ gelte. Spezielle Schnittebenenverfahren erhält man dann, indem man die Größen β_q und α_{qj} $(j \in I_N)$ definiert (x_q kann als (Typ 2)-Zeilenvariable aufgefaßt werden). Beispielsweise hat OWEN für Probleme (1) - (4), in denen nur (Typ 2)-Variablen auftreten (also $b_j^- = 0$, $b_j^+ = +\infty$, $N_k^+ = \emptyset$ für $j \in N$ und $k \in K$) die folgende Konstruktion vorgeschlagen (siehe [Owen, 1973]) [1)]

---

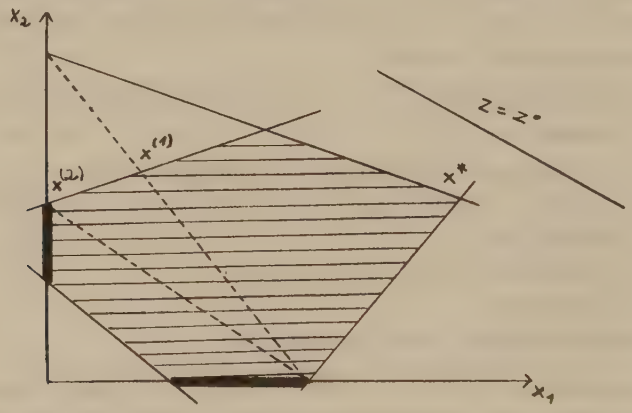
1) Zu ähnlichen Resultaten gelangten auch [Glover, 1973], [Young, 1971] und [Ibaraki, 1973].

$$\beta_q = -1$$

$$\alpha_{qj} = -\max_{i \in N_{k_o}} \frac{\alpha_{ij}}{\beta_i}, \quad j \in I_N$$

Dabei ist $k_o \in K$ eine disjunktive Restriktion, die durch die entsprechende Basislösung verletzt wird.

Die obige Konstruktion wurde später von BALAS und JEROSLOW als Spezialfall aus einem allgemeinen disjunktiven [1] Schnittprinzip abgeleitet und bewiesen (siehe [Balas, 1977], [Jeroslow, 1977]. Sie läßt sich geometrisch mit Hilfe der folgenden Abbildung veranschaulichen. [2]



Der schraffierte Bereich stellt die Menge der zulässigen Lösungen eines Systems (1) - (3) im $\mathbb{R}^2$ dar, dessen Optimum in x* erreicht wird. Die zusätzliche disjunktive Restriktion

$$x_1 \cdot x_2 = 0$$

führt zu den beiden dick gezeichneten Bereichen auf den x-Achsen. Verlängert man nun ausgehend von x* die Kanten des schraffierten Bereiches solange bis sie auf die Hyperebenen $x_1 = 0$ und

---

1) Die Vokabel "disjunktiv" wird dabei allerdings in einem anderen Sinne als in dieser Arbeit verwendet.

2) Vergl. [Owen, 1973].

$x_2 = 0$ treffen, so erhält man als Verbindungslinie dieser Schnittpunkte die erste Schnittebene, die gestrichelt gezeichnet ist und als neues Optimum des auf diese Weise verkleinerten schraffierten Bereichs $x^{(1)}$ liefert. Geht man nun analog von $x^{(1)}$ aus, so erhält man die zweite Schnittebene und die disjunktive optimale Lösung $x^{(2)}$.

Es soll nun gezeigt werden:

Satz: Für ein System der Form (1) - (4) liefert die Konstruktion

$$\beta_q = -1$$

$$\alpha_{qj} = \begin{cases} -\max(\vartheta_1, \vartheta_2, \vartheta_3), & \text{falls } j \notin I_N^{1*} \\ \max(\vartheta_4, \vartheta_5, \vartheta_6), & \text{falls } j \in I_N^{1*} \end{cases} \qquad (10)$$

mit

$$\vartheta_1 = \max_{i \in N_{k_o}^- \cap I_B} \frac{\alpha_{ij}}{|\beta_i|}, \quad \vartheta_2 = \max_{i \in N_{k_o}^+ \cap I_B} \frac{-\alpha_{ij}}{b_i^+ - \beta_i}, \quad \vartheta_3 = \begin{cases} 0, & \text{falls } (N_{k_o}^- \cup N_{k_o}^+) \cap (I_N \setminus \{j\}) \neq \emptyset \\ \frac{1}{b_j^+}, & \text{falls } j \in N_{k_o}^+ \end{cases}$$

$$\vartheta_4 = \max_{i \in N_{k_o}^- \cap I_B} \frac{-\alpha_{ij}}{|\beta_i|}, \quad \vartheta_5 = \max_{i \in N_{k_o}^+ \cap I_B} \frac{\alpha_{ij}}{b_i^+ - \beta_i}, \quad \vartheta_6 = \begin{cases} 0, & \text{falls } (N_{k_o}^- \cup N_{k_o}^+) \cap (I_N \setminus \{j\}) \neq \emptyset \\ \frac{1}{b_j^+}, & \text{falls } j \in N_{k_o}^- \end{cases}$$

wobei $k_o \in K$ eine durch die entsprechende Basislösung verletzte disjunktive Restriktion bezeichne, eine Schnittebene.

Beweis: Sei (2) und (3) in der Form

$$Ax = a$$
$$b^- \leq x \leq b^+$$

gegeben und seien die (Typ 1)-Variablen x_j zur Unterscheidung mit x_j^1 $(1 \leq 1 \leq 3)$ bezeichnet. Sei weiterhin

$$\bar{A}\,\bar{x} = \bar{a} \qquad (\bar{2})$$
$$\bar{x} \geq 0 \qquad (\bar{3})$$

dasjenige System, welches entsteht, wenn man in (2), (3) sämtliche (Typ 3)-Variablen x_j^3 durch die Differenz $x_j^+ - x_j^-$ mit $x_j^+, x_j^- \geq 0$ ersetzt und für alle (Typ 1)-Variablen x_j^1 Schlupfvariablen $x_j^s = b_j^+ - x_j^1$ mit $x_j^s \geq 0$ einführt.

Es gibt nun zu jeder Basislösung x von (2),(3) mit der Basis B eine Basislösung $\bar{x}$ von $(\bar{2}),(\bar{3})$ mit der Basis $\bar{B}$, so daß - eventuell nach Umnumerierung der Indizes - die folgenden Beziehungen gelten: [1]

$$\bar{\alpha}_j^{\bar{1}}=\begin{pmatrix}\alpha_j\\-\alpha_j^1\\e_j^1\end{pmatrix},\quad \bar{\alpha}_j^{\bar{s}}=\begin{pmatrix}-\alpha_j\\\alpha_j^1\\e_j^s\end{pmatrix},\quad \bar{\alpha}_j^{\bar{2}}=\begin{pmatrix}\alpha_j\\-\alpha_j^1\\0\end{pmatrix},\quad \bar{\alpha}_j^{\bar{+}}=\begin{pmatrix}\alpha_j\\-\alpha_j^1\\0\end{pmatrix},\quad \bar{\alpha}_j^{\bar{-}}=\begin{pmatrix}-\alpha_j\\\alpha_j^1\\0\end{pmatrix},\quad \bar{\beta}=\begin{pmatrix}|\beta|\\b_{B1}^{+}-\beta^1\\b_{N1}^{+}\end{pmatrix}$$

Dabei ist

$\bar{\alpha}_j^{\bar{1}} = \bar{A}_{\bar{B}}^{-\bar{1}}\bar{A}_j^{\bar{1}}$, $j\in I_N$, $1\in\{1,s,2,+,-\}$,

$\bar{A}_j^{\bar{1}}$: Spalte in $\bar{A}$, die zu x_j^1 ($1\in\{1,s,2,+,-\}$) gehört,

$\alpha_j = A_B^{-1}A_j$, $j\in I_N$,

α_j^1 : Vektor der Komponenten α_{ij} von α_j mit $i\in I_B^1$,

e_j^1 : Einheitstupel, dessen Eins an der gleichen Stelle steht wie der Wert der Basisvariablen x_j^1 in $\bar{\beta}$ ($1\in\{1,s\}$),

$\bar{\beta} = \bar{A}_{\bar{B}}^{-\bar{1}}\bar{a}$,

$\beta = A_B^{-1}a - \sum_{j\in I_N^{1*}} b_j^{+}\alpha_j$

β^1 : Vektor der Komponenten β_i von β mit $i\in I_B^1$,

b_{B1}^{+}: Vektor der Komponenten b_i^{+} von b^{+} mit $i\in I_B^1$,

b_{N1}^{+}: Vektor der Komponenten b_j^{+} von b^{+} mit $j\in I_N^1$.

Aus den obigen Beziehungen folgt mittels Ausrechnung, daß die Anwendung der Konstruktion (10) auf (1) - (4) der Anwendung des OWEN-Cut auf $(\bar{1})$ - $(\bar{4})$ entspricht. Ihre Gültigkeit ergibt sich somit aus der Gültigkeit des OWEN-Cut.

□

Der soeben bewiesene Satz liefert zusammen mit den zu Beginn des Paragraphen vorgenommenen Spezifizierungen des allgemeinen

---

1) Zur ausführlichen Herleitung siehe [Hadley, 1962, S. 387 ff.]

Algorithmusschemas von Seite 57 ein Schnittebenenverfahren zur Lösung von Problemen der Form (1) - (4). Dieses soll nun an Hand des Zahlenbeispiels von Seite 8 illustriert werden.

Löst man zunächst (Π_R) = (1) - (3) mit Hilfe des Simplexverfahrens (vergl. S. 34), so erhält man das Tableau

			*		*		
			1	2	1	1	
			x_7	x_8	x_6	x_1	
			-12	5	-8	3	65
→	2	x_4	- 1	1	-1	0	6
	1	x_3	0	0	-1	-1	7
↓	1	x_5	- 1	1	-1	-1	6
	3	x_2	1	0	0	-1	-2

Zur Verdeutlichung der im obigen Beweis verwendeten Bezeichnungen sei auch das entsprechende Tableau für das System $(\bar{1})$ - $(\bar{3})$ angegeben.

		x_7^s	x_8^2	x_6^s	x_1^1	x_2^+		
		12	5	8	3	0	65	
→	x_4^2	1	1	1	0	0	6	} $\lvert\beta\rvert$
	x_3^1	0	0	1	-1	0	7	
	x_5^1	1	1	1	-1	0	6	
	x_2^-	-1	0	0	-1	-1	2	
	x_3^s	0	0	-1	1	0	1	} $b_{B1}^+ - \beta^1$
→	x_5^s	-1	-1	-1	1	0	1	
	x_7^1	1	0	0	0	0	3	} b_{N1}^+
	x_6^1	0	0	1	0	0	2	
	x_1^s	0	0	0	1	0	2	

Die Anwendung des OWEN-Cut bzw. der Konstruktion (10) mit $k_o = 1$ (die betreffenden Indizes i sind mit $\rightarrow$ bzw. mit $\bar{\rightarrow}$, falls $i \in N^-_{k_o}$, und mit $\overset{+}{\rightarrow}$, falls $i \in N^+_{k_o}$, gekennzeichnet) ergibt somit die zusätzliche Zeile

x^2_9	$-\frac{1}{6}$	$-\frac{1}{6}$	$-\frac{1}{6}$	-1	0	-1

bzw.

2	x_9	$\frac{1}{6}$	$-\frac{1}{6}$	$\frac{1}{6}$	(-1)	-1

Für (1) - (4) seien noch die weiteren im Verlauf des Verfahrens auftretenden Tableaus aufgeführt: (Die Pivotelemente sind jeweils eingekreist.)

			*		*		
			1	2	1	2	
			x_7	x_8	x_6	x_9	
			-11,5	4,5	-7,5	3	62
	2	x_4	- 1	1	-1	0	6
$\bar{\rightarrow}$	1	x_3	- 0,17	0,17	-1,17	-1	8
	1	x_5	- 1,17	1,17	-1,17	-1	7
	3	x_2	0,83	0,17	-0,17	-1	-1
$\overset{+}{\rightarrow}$	1	x_1	- 0,17	0,17	-0,17	-1	1
	2	x_{10}	0,02	-0,02	0,15	(-1)	-1

		*		*		
		1	2	1	2	
		x_7	x_8	x_6	x_{10}	
		-11,44	4,44	-7,05	3	59
2	x_4	- 1	1	-1	0	6
1	x_3	- 0,19	0,19	(-1,32)	-1	9
1	x_5	- 1,19	1,19	-1,32	-1	8
3	x_2	0,81	0,19	-0,32	-1	0
1	x_1	- 0,19	0,19	-0,32	-1	2
2	x_9	- 0,02	0,02	-0,15	-1	1

		*		↓ *		
		1	2	1	2	
		x_7	x_8	x_3	x_{10}	
		-10,43	3,43	-5,34	8,34	53,64
2	x_4	- 0,86	0,86	-0,76	0,76	5,24
1	x_6	0,14	-0,14	-0,76	0,76	1,24
1	x_5	- 1	1	-1	0	7
3	x_2	0,86	0,14	-0,24	-0,76	-0,24
1	x_1	- 0,14	0,14	-0,24	-0,76	1,76
2	x_9	0	0	-0,11	-0,89	0,89
2	x_{11}	0	0	0,125	(-3,17)	-1

←

			*		*	
		1	2	1	2	
		x_7	x_8	x_3	x_{11}	
		-10,43	3,43	-5	2,63	51
2	x_4	- 0,86	0,86	-0,73	0,24	5
1	x_6	0,14	-0,14	-0,73	0,24	1
1	x_5	- 1	1	-1	0	7
3	x_2	0,86	0,14	-0,27	-0,24	0
1	x_1	- 0,14	0,14	-0,27	-0,24	2
2	x_9	0	0	-0,15	-0,28	1,17
2	x_{10}	0	0	-0,39	-0,32	0,32

Die dem letzten Tableau entsprechende Basislösung ist zulässig und damit optimal.

7. Branch-and-Bound-Verfahren

Im Gegensatz zu den Schnittebenenverfahren, die nur das Prinzip der Relaxation und beim Verwerfen den Fall (c1) bzw. (c3) (am Ende) berücksichtigen, verwenden Branch-and-Bound-Verfahren in erster Linie das Prinzip der Separation (auch "Branching" genannt) und beim Verwerfen den Fall (c2) (Berechnung der "Bounds"). Das allgemeine Algorithmusschema erfährt dabei folgende Spezifizierungen:

4. Relaxation der Disjunktivitätsbedingungen.
10. Übergang stets zu Schritt 11.
12. Entfällt.

Unterschiede zwischen den einzelnen Branch-and-Bound-Verfahren ergeben sich vor allem durch die Art und Weise der Auswahl bzw. Separation des Kandidatenproblems $(\Pi) \in C$ in Schritt 3 bzw. Schritt 11. Hierzu werden heute fast ausschließlich Strafkosten (auch Pseudokosten; engl.: penalties, pseudo-costs) verwendet. Sie entstehen dadurch, daß man für ein Kandidatenproblem keine neue obere Schranke durch eine vollständige Simplexiteration berechnet, sondern abschätzt, wie stark sich der Zielwert bei Übergang zu einer disjunktiven Lösung mindestens ändern muß. Dies soll an Hand des Zahlenbeispiels erläutert werden. Ausgangspunkt ist wiederum das optimale Tableau für (1) - (3):

			*		*	
			1	2	1	1
			x_7	x_8	x_6	x_1	
			-12	5	-8	3	65
2	x_4	- 1	1	-1	0	6
1	x_3	0	0	-1	-1	7
1	x_5	- 1	1	-1	-1	6
3	x_2	1	0	0	-1	-2

Legt man bzgl. einer linear zulässigen jedoch nichtdisjunktiven Basislösung x von (1) - (4) als Strafkosten $p(x_i)$ der in einer verletzten disjunktiven Restriktion $k_o \in K$ auftretenden Variablen x_i den Betrag fest, um den der Zielwert abnimmt, wenn man x_i mit Hilfe eines dualen Simplexschrittes auf einen für k_o relevanten Rand setzt, so erhält man aus dem obigen Tableau wegen

$$(7 - x_5)\cdot x_4 \neq 0 \qquad (k_o = 1),$$

$$p(x_4) = 30 ,$$

$$p(x_5) = 3 ,$$

und wegen

$$(2 - x_1)\cdot x_3 \neq 0 \qquad (k_o = 2),$$

$$p(x_1) = 6 ,$$

$$p(x_3) = 56 .$$

Formalisiert ergibt sich:

$$p(x_i) = \begin{cases} \alpha_{oi}\, b_i^+, & \text{falls } i \in N_{k_o}^+ \cap I_N \\ -\alpha_{oi}\, b_i^+, & \text{falls } i \in N_{k_o}^- \cap I_N \\ (\beta_i - b_i^+)\cdot \min(\sigma_1,\sigma_2), & \text{falls } i \in N_{k_o}^+ \cap I_B \text{ und } \min(\sigma_1,\sigma_2) \text{ existiert} \\ \infty, & \text{falls } i \in N_{k_o}^+ \cap I_B \text{ und } \min(\sigma_1,\sigma_2) \text{ nicht existiert} \\ \beta_i \cdot \min(\sigma_3,\sigma_4), & \text{falls } i \in N_{k_o}^- \cap I_B \text{ und } \min(\sigma_3,\sigma_4) \text{ existiert} \\ \infty, & \text{falls } i \in N_{k_o}^- \cap I_B \text{ und } \min(\sigma_3,\sigma_4) \text{ nicht existiert} \end{cases}$$

Dabei ist σ_l $(1 \leq l \leq 4)$ wie auf Seite 28 definiert. Die Validität dieser Strafkostendefinition, die derjenigen von DRIEBEEK für 0/1-Optimierungsprobleme (siehe [Driebeek, 1966]) und von IBARAKI für lineare Komplementärprobleme (siehe [Ibaraki, 1971]) ähnelt, wird aus dem folgenden Satz ersichtlich:

Satz: Sei (Π) ein Kandidatenproblem von (P) und $x \notin F(P)$ eine optimale Basislösung von (Π_R).
Ist für eine Variable x_i

$$\underbrace{z^*(\Pi_R)}_{cx} - p(x_i) \leqq \hat{z}, \qquad (11)$$

dann gilt

$$z^*(\Pi_i) \leqq z^*(P),$$

wobei (Π_i) dasjenige Problem ist, welches entsteht, wenn man zu (Π) die Restriktion

$$x_i = 0 \quad , \text{ falls } i \in N_{k_o}^-,$$

oder

$$x_i = b_i^+ \quad , \text{ falls } i \in N_{k_o}^+,$$

hinzufügt.

Beweis: Offensichtlich ist

$$z^*(\Pi_i) \leqq z^*(\Pi_R) - p(x_i) \underset{(n.V.)}{\leqq} \hat{z} \leqq z^*(P)$$

□

Da die Gestalt von (P) es nahelegt, die Separation so durchzuführen, daß man nach und nach disjunktive Variablen auf einen relevanten Rand fixiert, besagt der vorangegangene Satz, daß ein Kandidatenproblem (Π_i) unmittelbar gemäß (c2) verworfen werden kann, wenn (11) erfüllt ist. Das zugehörige relaxierte Problem braucht somit nur dann mit Hilfe der Simplexmethode gelöst zu werden (Schritt 5), wenn (11) nicht erfüllt ist. Auf diese Weise führt die Verwendung von Strafkosten zu einer beträchtlichen Rechenzeitverbesserung.

Bevor das Verfahren an Hand des Zahlenbeispiels illustriert werden soll, seien noch die Schritte 11 und 3 wie folgt präzisiert:

11. Separation von (Π) in $(\Pi_{j_1}),\ldots,(\Pi_{j_L})$ und Neufestsetzung von $C := C \cup \{(\Pi_{j_1}),\ldots,(\Pi_{j_L})\}$, wobei (Π_{j_1}) $(1 \leqq l \leqq L)$ dasjenige System sei, welches entsteht, wenn man zu (Π)

die Restriktion

$$x_{j_1} = 0 \quad , \quad \text{falls } j_1 \in N^-_{k_o} \ ,$$

oder

$$x_{j_1} = b^+_{j_1} \quad , \quad \text{falls } j_1 \in N^+_{k_o} \ ,$$

hinzufügt. Dabei sei $L = |N^-_{k_o} \cup N^+_{k_o}|$ und $k_o \in K$ eine von der optimalen Basislösung x von (Π_R) verletzte disjunktive Restriktion, für die gilt:

$$|p(x_{i_{k_o}}) - \min_{i \in N^-_{k_o} \cup N^+_{k_o}} p(x_i)| = \max_{k \in K(x)} |p(x_{i_k}) - \min_{i \in N^-_k \cup N^+_k} p(x_i)|,$$

$$i \neq i_{k_o} \qquad\qquad i \neq i_k$$

mit $$p(x_{i_k}) = \min_{i \in N^-_k \cup N^+_k} p(x_i)$$

und

$K(x)$: Menge der von der optimalen Basislösung x von (Π_R) verletzten disjunktiven Restriktionen.

3. Wahl von

$$(\Pi) := \begin{cases} (\Pi_{i_{k_o}}) \ , \text{ falls } K(x) \neq \emptyset \\ \\ (\Pi'), \quad \text{andernfalls} \end{cases}$$

und Neufestsetzung von $C := C \setminus \{(\Pi)\}$, wobei i_{k_o} in Schritt 11 der vorausgegangenen Iteration festgelegt wurde und (Π') das zuletzt in C aufgenommene System ist.

Schritt 11 besagt, daß (Π) nach derjenigen in einem optimalen Simplextableau von (Π_R) (welches in Schritt 5 ermittelt wurde) verletzten disjunktiven Restriktion $k_o \in K$ separiert wird, für die die Differenz der entsprechenden kleinsten und zweitkleinsten Strafkosten betragsmäßig maximal ist.

In Schritt 3 der nächsten Iteration wird dann das zu den kleinsten Strafkosten (bzgl. k_o) gehörende System $(\Pi_{i_{k_o}})$ ausgewählt.

Der so definierten Separations- und Auswahlregel liegt die folgende Überlegung zugrunde: Wenn innerhalb einer verletzten disjunktiven Restriktion $k_o \in K$ die Fixierung einer bestimmten Variablen $x_{i_{k_o}}$ auf einen relevanten Rand soviel weniger "Strafe kostet" als die Fixierung einer der übrigen Variablen auf einen relevanten Rand, dann ist die Wahrscheinlichkeit groß, daß $x_{i_{k_o}}$ auch in einer optimalen Lösung von (P) auf einem relevanten Rand liegt. Es ist daher sinnvoll, $(\Pi_{i_{k_o}})$ als Kandidatenproblem zu wählen. Wird in einer Iteration Schritt 11 nicht durchlaufen (d.h. $K(x) = \emptyset$ oder Π_R ist unlösbar), so wird die Kandidatenliste C weiter abgearbeitet ($(\Pi) := (\Pi')$).

Aus dem Simplextableau von Seite 67 erhält man also, wegen

$|p(x_1)-p(x_3)| = 50 > |p(x_4)-p(x_5)| = 27$,
$(\Pi) = (\Pi_1)$ und $C = \{(\Pi_3)\}$.

Eine Lösung von (Π_{1R}) ergibt die Tableaus

			*		*	[*]
		1	2	1	1	
		x_7	x_8	x_6	x_1	
		-12	5	-8	3	59
2	x_4	- 1	1	-1	0	6
1	x_3	0	0	(-1)	-1	9
1	x_5	- 1	1	-1	-1	8
3	x_2	1	0	0	-1	0

		*		*	[*]	
		1	2	1	1	
		x_7	x_8	x_3	x_1	
		-12	5	-8	11	51
2	x_4	- 1	1	-1	1	5
1	x_6	0	0	-1	1	1
1	x_5	- 1	1	-1	0	7
3	x_2	1	0	0	-1	0

Da nun $K(x) = \emptyset$ ist, wird $\hat{z} = 51$ und $(\Pi) = (\Pi_3)$ gesetzt. Wegen $65 - 56 \not< 51$, ist (11) erfüllt und (Π_3) kann daher unmittelbar verworfen werden. Da nun $C = \emptyset$ ist, liefert das letzte Tableau bereits eine optimale Lösung von (1) - (4).

8. Kombination von Schnittebenenverfahren und Branch-and-Bound-Verfahren.

Eine Kombination der in den beiden vorangegangenen Paragraphen beschriebenen Spezifizierungen des allgemeinen Algorithmusschemas erhält man, wenn man nach der Abfrage in Schritt 10 sowohl den Übergang zu Schritt 11 als auch zu Schritt 12 zuläßt. Schritt 10 kann dabei beispielsweise folgendermaßen präzisiert werden:

10. Abfrage ob

$$\left| \frac{z^*(\Pi_R^{q-1}) - z^*(\Pi_R^q)}{z^*(\Pi_R^{q-1})} \right| > \varepsilon$$

gilt.

Schritt 10 besagt, daß das Kandidatenproblem (Π) erst dann separiert wird, wenn sich die Einführung weiterer Schnittebenen nicht mehr lohnt (q ist die Anzahl der bereits zu (Π) hinzugefügten Schnittebenen). Als Kriterium hierfür wird die relative Veränderung des Zielwertes herangezogen. Unterschreitet diese eine vorgegebene untere Schranke ε (z.B. $\varepsilon = 0{,}001$), wird zu Schritt 12 übergegangen, d.h. (Π) wird weiter separiert.

Wegen

$$z^*(\Pi_R^q) \leq z^*(\Pi_R)$$

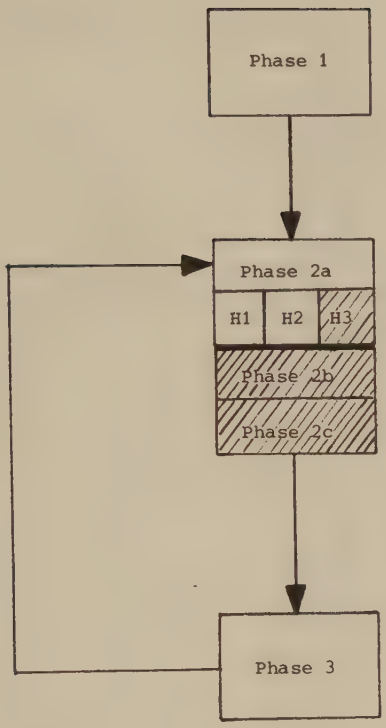
führt diese Vorgehensweise gegenüber dem Branch-and-Bound-Verfahren aus § 7 zu einer weiteren Verbesserung des Verwerfkriteriums (c2), d.h. die Anzahl der innerhalb des Verfahrens zu lösenden Teilsysteme wird verringert, allerdings auf Kosten des zusätzlichen Aufwands für die Berechnung der Schnittebenen.

9. Effizientes Lösungsverfahren

Die in den vorangegangenen Paragraphen beschriebenen, aus der ganzzahligen Optimierung abgeleiteten Verfahren, enden zwar stets mit einer optimalen Lösung von (1) - (4), falls eine solche existiert, sind in der Regel jedoch nur auf kleine bis mittelgroße Probleme anwendbar, da ihr Rechenaufwand überaus schnell mit der Anzahl der disjunktiven Restriktionen ansteigt.

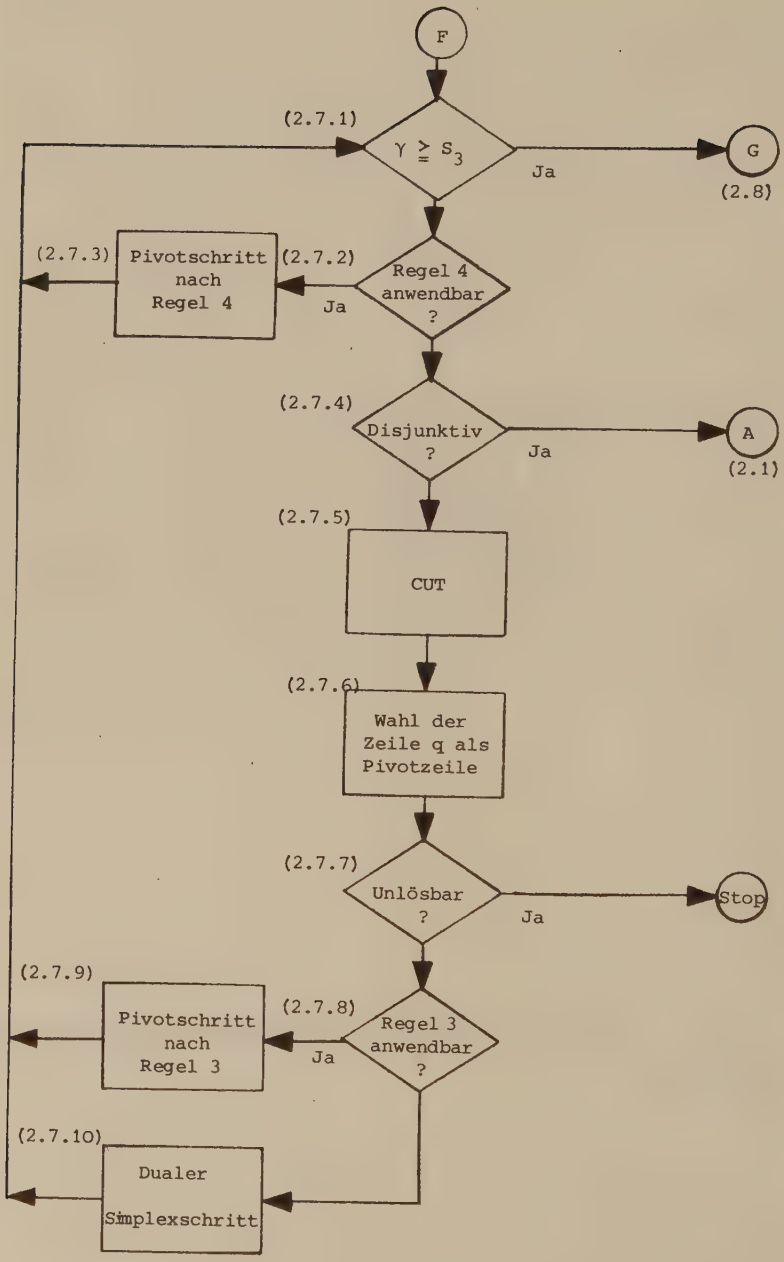
Im Gegensatz zu der im § 4 vorgestellten erweiterten Simplexmethode liegt das Dilemma für den Benutzer hier nicht in der Lösungsqualität, sondern in der Rechenzeit. Als Ausweg aus diesem Dilemma bietet sich die Kombination beider Ansätze an. Dazu werden in die zweite Phase der erweiterten Simplexmethode die Lösungsprinzipien Relaxation, Separation und Verwerfen so eingebaut, daß das Nichtauffinden einer zulässigen Lösung von (1) - (4) mit größerem Zielwert als $\hat{z} + \lambda$ gleichbedeutend mit der Nichtexistenz einer solchen Lösung ist und somit dann die Abschätzung (9) auf Seite 42 gilt. Das auf diese Weise entstehende neue Verfahren zur Lösung von (1) - (4) läßt sich durch das folgende Flußdiagramm veranschaulichen (die schraffierten Felder zeigen dabei den Unterschied zu dem erweiterten Simplexverfahren aus § 4):

Verfahren

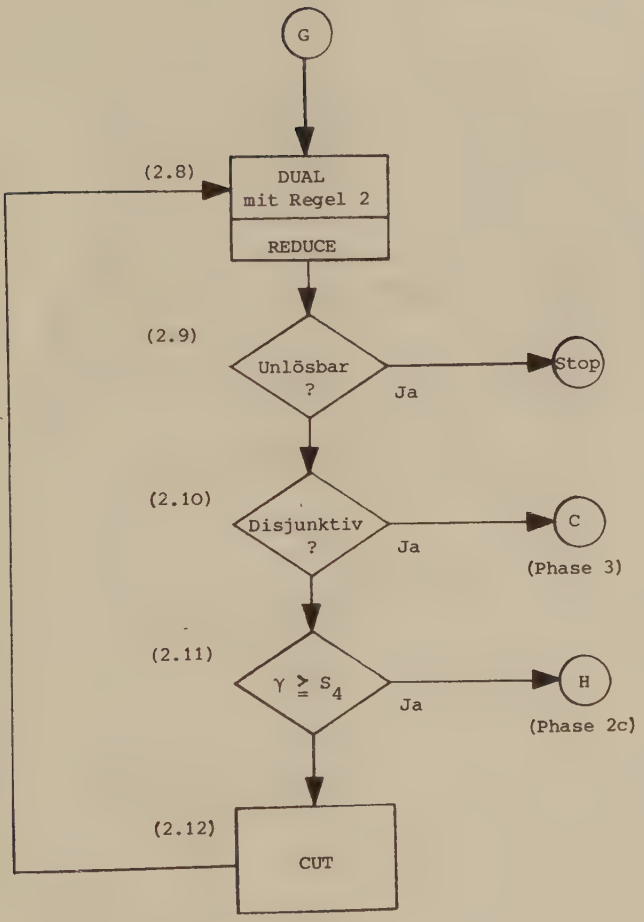


Heuristik 3 (H3), Phase 2b und Phase 2c haben dabei das folgende Aussehen:

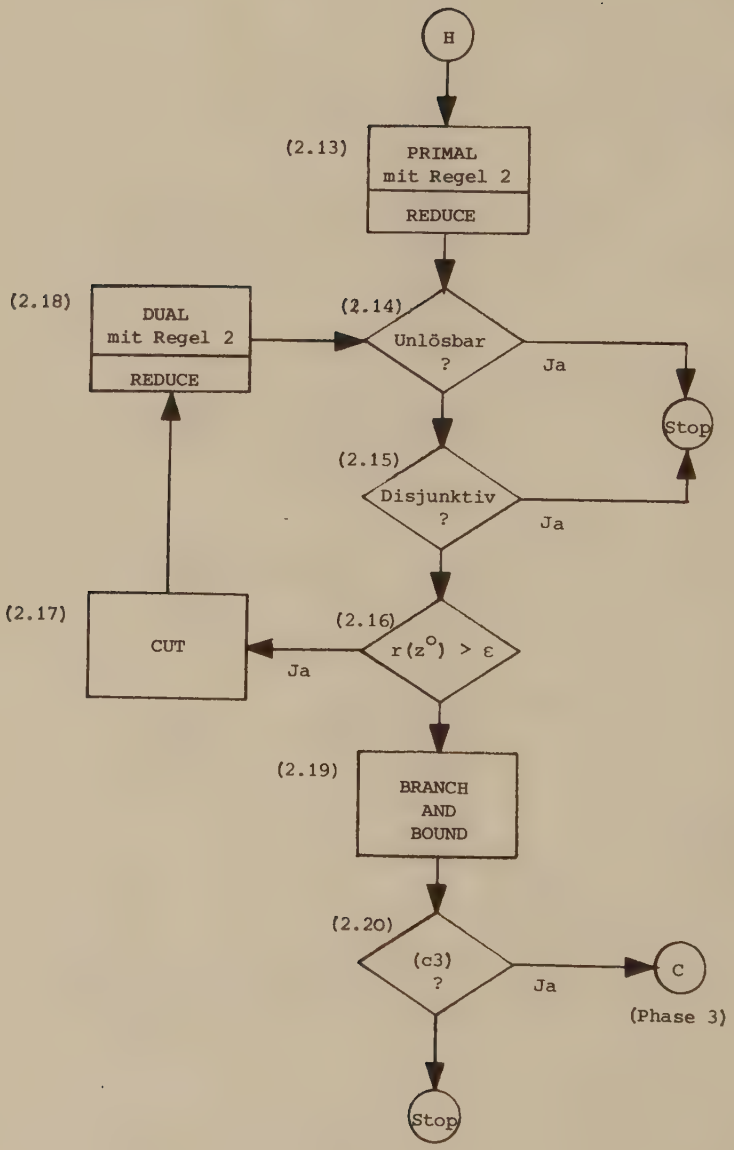
Heuristik 3


F
(2.7.1)
$\gamma \geq s_3$
Ja
G
(2.8)
(2.7.3)
Pivotschritt nach Regel 4
(2.7.2)
Regel 4 anwendbar ?
Ja
(2.7.4)
Disjunktiv ?
Ja
A
(2.1)
(2.7.5)
CUT
(2.7.6)
Wahl der Zeile q als Pivotzeile
(2.7.7)
Unlösbar ?
Ja
Stop
(2.7.9)
Pivotschritt nach Regel 3
(2.7.8)
Regel 3 anwendbar ?
Ja
(2.7.10)
Dualer Simplexschritt

Phase 2b


G
(2.8)
DUAL
mit Regel 2
REDUCE
(2.9)
Unlösbar
?
Ja
Stop
(2.10)
Disjunktiv
?
Ja
C
(Phase 3)
(2.11)
$\gamma \geqq S_4$
Ja
H
(Phase 2c)
(2.12)
CUT

Phase 2c


H
(2.13)
PRIMAL
mit Regel 2
REDUCE
(2.18)
DUAL
mit Regel 2
REDUCE
(2.14)
Unlösbar
?
Ja
Stop
(2.15)
Disjunktiv
?
Ja
(2.17)
CUT
(2.16)
r(z°) > ε
Ja
(2.19)
BRANCH
AND
BOUND
(2.20)
(c3)
?
Ja
C
(Phase 3)
Stop

Erläuterungen

Phase 1: Siehe Flußdiagramm auf Seite 46 nebst Erläuterungen (1.1) - (1.3) auf Seite 51.

Phase 2a: Siehe Flußdiagramme auf Seite 47 - 49 nebst Erläuterungen (2.1) - (2.6) auf Seite 51 und 52. Da nun drei heuristische Strategien durchlaufen werden, tritt S_3 in (2.4) an die Stelle von S_2. Weiterhin treten an die Stelle von (Stop) nach der Abfrage von (2.4) bzw. in Heuristik 2 die Marken (G) bzw. (F), um den Anschluß zur Phase 2b bzw. Heuristik 3 herzustellen.

(2.7) Heuristik 3 dient wie Heuristik 2 der Wiederherstellung der Disjunktivität. Allerdings wird hierbei die Konstruktion (10) von Seite 61 verwendet, um zusätzlich den Lösungsbereich möglichst rasch zu verkleinern.

(2.7.1) Abfrage nach Überschreitung der Zeit- oder Iterationsschranke etc. für das Durchlaufen der Heuristik 3 ($S_1 \leqq S_2 \leqq S_3$).

(2.7.2) Abfrage, ob ein Pivotschritt unter Beachtung der Regel 4 (siehe S. 39) möglich ist.

(2.7.3) Pivotschritt unter Beachtung der Regel 4.

(2.7.4) Abfrage nach der Disjunktivität der vorliegenden Lösung.

(2.7.5) Einführung einer zusätzlichen Restriktion gemäß der Konstruktion (10) von Seite 61 (CUT). Man beachte, daß das hierbei zugrundeliegende Simplextableau in der Regel linear unzulässig ist, d.h. nicht mehr unbedingt von einer Schnittebene im Sinne der Definition auf Seite 59 gesprochen werden kann. Wie unten gezeigt wird, ist jedoch auch in diesem Fall gesichert, daß CUT nur nichtdisjunktive Lösungen abtrennt, ist es aber möglich, daß sowohl besonders tiefe als auch wirkungslose Schnitte erzeugt werden.

(2.7.6) Wahl der hinzugefügten Zeile q als Pivotzeile.

(2.7.7) Abfrage, ob die vorliegende Basis auf die Nichtexistenz einer zulässigen Lösung hinweist (vergl. (2.3) auf S.28).

(2.7.8) Abfrage, ob ein Pivotschritt mit der Zeile q als Pivotzeile unter Beachtung der Regel 3 (siehe S. 38) möglich ist.

(2.7.9) Pivotschritt mit der Zeile q als Pivotzeile unter Beachtung der Regel 3.

(2.7.10) Durchführung eines dualen Simplexschrittes mit der Zeile q als Pivotzeile.

Phase 2b:

(2.8) Anwendung von DUAL (siehe S. 24) unter Beachtung der Regel 2 (siehe S. 38) bis zum Abbruch. Um eine zu große Aufblähung der Arbeitsmatrix durch das Hinzufügen der Schnittebenen in (2.12) zu verhindern, sollte zusätzlich die Prozedur REDUCE (siehe Abschnitt 4.1.3) benutzt werden. Diese wird dabei um die folgende Vorgehensweise erweitert: Wenn bez. einer disjunktiven Restriktion $k_o \in K$ eine Schnittebene eingeführt wird, dann werden alle bereits bez. k_o erzeugten Schnittebenen eliminiert, sobald sich die entsprechenden Variablen in der Basis befinden, so daß pro disjunktive Restriktion möglichst nur eine Schnittebene mitgeführt wird. Diese Vorgehensweise ist gerechtfertigt, da - wie man sich leicht an Hand der geometrischen Interpretation (siehe S. 60 und S. 83) verdeutlicht - durch die Einführung einer neuen Schnittebene bez. k_o in den meisten Fällen vorher bez. k_o erzeugte Schnittebenen überflüssig werden.

(2.9) Abfrage, ob die vorliegende Basis auf die Nichtexistenz einer zulässigen Lösung hinweist.

(2.10) Abfrage, ob die in (2.8) berechnete linear zulässige Basislösung disjunktiv ist.

(2.11) Abfrage nach Überschreitung der Zeit- oder Iterationsschranke für das Durchlaufen der Phase 2b ($S_1 \leq S_2 \leq S_3 \leq S_4$).

(2.12) Einführung einer Schnittebene gemäß der Konstruktion (10) von Seite 61.

Phase 2c:

(2.13) Anwendung von PRIMAL (siehe S. 25) unter Beachtung der Regel 2 bis zum Abbruch und zusätzlich Benutzung der Prozedur REDUCE.

(2.14) Abfrage, ob die vorliegende Basis auf die Nichtexistenz einer zulässigen Lösung hinweist.

(2.15) Abfrage, ob die in (2.13) berechnete linear optimale Basislösung disjunktiv ist.

(2.16) Abfrage, ob die Einführung weiterer Schnittebenen noch sinnvoll ist. Dabei ist

$$r(z^o) = \left| \frac{\tilde{z} - z^o}{\tilde{z}} \right| , \; z^o \; (\tilde{z})$$

der Zielwert der (vor)letzten ermittelten Basislösung und ε eine vorgegebene untere Schranke (vergl. § 8, S. 73). Wird also die relative Veränderung des Zielwertes zu gering, d.h. erscheint die Einführung weiterer Schnittebenen wirkungslos, geht man zu (2.19) über.

(2.17) Einführung einer Schnittebene gemäß der Konstruktion (10) von Seite 61.

(2.18) Anwendung von DUAL unter Beachtung der Regel 2 bis zum Abbruch und zusätzlich Benutzung der Prodzedur REDUCE (vergl. (2.8) in Phase 2b).

(2.19) Anwendung des in § 7 geschilderten Branch-and-Bound-Verfahrens mit dem Unterschied, daß das Verfahren auch dann als beendet gilt, sobald der Fall (c3) (siehe S. 56) eintritt (BRANCH AND BOUND). Dabei wird in Schritt 1 (siehe S. 58) $C := \{(\Pi)\}$ und $\hat{z} := \hat{z} + \lambda$ gesetzt, wobei (Π_R) dasjenige System ist, für welches das letzte aus (2.18) vorliegende Simplextableau eine optimale Basislösung liefert.

(2.20) Abfrage ob BRANCH AND BOUND mit dem Fall (c3) endet, d.h. mit einer zulässigen Lösung von (1) - (4) die (9) erfüllt.

Phase 3: Siehe Flußdiagramm auf Seite 50 nebst Erläuterungen (3.1) - (3.9) auf Seite 52 - 53.

Bleibt noch zu zeigen:

Satz: Das soeben dargestellte Verfahren zur Lösung von (1) - (4) führt nach endlich vielen Schritten zu einer zulässigen Lösung, für deren Zielwert (9) erfüllt ist oder zu der Aussage, daß keine zulässige Lösung existiert.

Dazu sei zunächst bewiesen:

Lemma: Durch die Prozedur CUT werden stets nur Bereiche nichtdisjunktiver Lösungen abgetrennt.

Beweis:

Fall 1: Das zugrundeliegende Simplextableau ist linear zulässig. Das Lemma folgt unmittelbar aus dem Satz auf Seite 61, da durch CUT eine Schnittebene erzeugt wird.

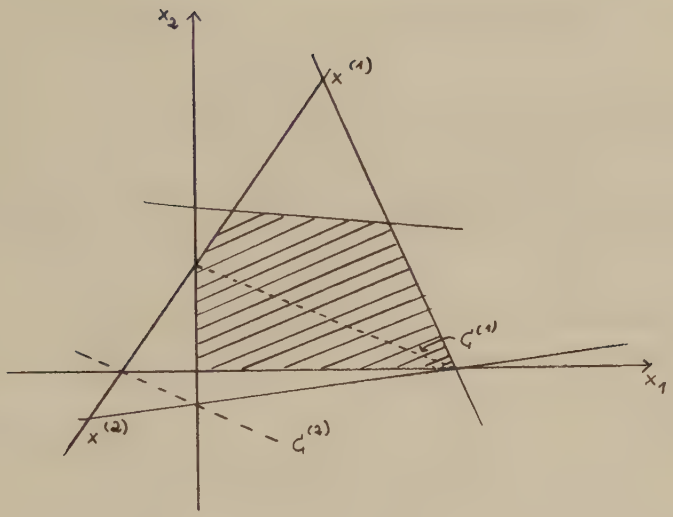
Fall 2: Das zugrundeliegende Simplextableau ist linear unzulässig (siehe Heuristik 3).

Unterfall 2a: Es treten keine (Typ 1)-Variablen auf.
Durch Freigabe der (Typ 2)-Basisvariablen mit negativen Werten (Typänderung auf Typ 3) erhält man ein linear zulässiges Simplextableau für ein System mit vergrößertem linear zulässigen Lösungsbereich. CUT liefert für beide Systeme dieselbe zusätzliche Restriktion und trennt von dem linear zulässigen Bereich des relaxierten Systems nur nichtdisjunktive Lösungen ab (Fall 1), woraus die Gültigkeit des Lemmas unmittelbar folgt.

Unterfall 2b: Es treten (Typ 1)-Variablen auf.
Durch Einführung von Schlupfvariablen läßt sich dieser Fall unmittelbar auf Unterfall 2a zurückführen.

□

Wie bereits erwähnt wurde, kann CUT ausgehend von einer linear unzulässigen Basislösung besonders tiefe Schnitte liefern (worauf in Heuristik 3 spekuliert wird), hinsichtlich der Verkleinerung des linear zulässigen Lösungsbereichs jedoch auch wirkungslos bleiben. Dies läßt sich leicht an Hand der folgenden Abbildung illustrieren (vergl. S. 60):



Der linear zulässige Lösungsbereich sei schraffiert. Ausgehend von $x^{(1)}$ erzeugt CUT bez. der disjunktiven Restriktion $x_1 \cdot x_2 = 0$ den Schnitt $C^{(1)}$, der den schraffierten Bereich mehr verkleinert als sämtliche entsprechende Schnitte, denen linear zulässige Basislösungen zugrunde liegen. Ausgehend von $x^{(2)}$ erzeugt CUT bez. der disjunktiven Restriktion $x_1 \cdot x_2 = 0$ den Schnitt $C^{(2)}$, der den schraffierten Bereich nicht verkleinert.

Beweis des Satzes:

Da das Simplexverfahren endlich ist (theoretisch mögliches "Kreisen" läßt sich mit Hilfe von Störmethoden verhindern) [1] und innerhalb des Branch-and-Bound-Verfahrens in (2.19) nur eine endliche Anzahl von Kandidatenproblemen auftreten kann (vergl. S. 10), ansonsten die Iterationsanzahl vom Benutzer durch Schranken S_i ($1 \leqq i \leqq 4$) und ε begrenzt wird, endet das Verfahren nach endlich vielen Schritten. Da durch die Prozedur CUT bzw. JUMP nur Bereiche von nichtdisjunktiven Lösungen bzw. von Lösungen, deren Zielwerte die rechte Seite von (9) $\hat{z} + \hat{\lambda}$ nicht überschreiten, abgetrennt werden, können bei Abbruch des Verfahrens nur die folgenden Fälle vorliegen:

Fall 1: Der (reduzierte) Lösungsbereich von (1) - (4) ist leer. Wie bereits erläutert, ist dieser Fall dann gegeben, wenn in DUAL σ nicht existiert (siehe (2.3), (2.7.7), (2.9), (2.14)), REDUCE einen Widerspruch liefert (siehe (2.9), (2.14), (3.5)) oder die Kandidatenliste C in BRANCH AND BOUND leer ist (siehe (2.20)). Liegt eine gespeicherte Lösung vor, so genügt deren Zielwert der Abschätzung (9). Andernfalls besitzt (1) - (4) keine zulässige Lösung.

Fall 2: Die vorliegende linear optimale Basislösung ist disjunktiv (siehe (2.15), (3.3)).
In diesem Fall ist $\hat{z} = z^*$ und (9) somit erfüllt.

Fall 3: Der Zielwert von (1) - (4) ist unbeschränkt. Dieser Fall ist dann gegeben, wenn in PRIMAL unter Beachtung der Regel 1 δ nicht existiert (siehe 3.4). Es gilt dann $\hat{z} = z^* = +\infty$ und somit (9).

□

Zum Abschluß dieses Paragraphen sei noch erwähnt, daß das vorgestellte Verfahren überaus flexibel bei der Anpassung an Spezialprobleme der Form (1) - (4) ist. Dies sei am Beispiel der linea-

---

1) Siehe Fußnote 1) auf Seite 31.

ren Optimierungsprobleme mit zusätzlicher kardinaler Beschränktheitsbedingung

$$\begin{aligned} cx &\to \max! \\ Ax &= a \\ b^- \leqq x &\leqq b^+ \\ |\{x_j > 0 : j\in J\}| &\leqq q \end{aligned}$$

demonstriert (vergl. 2.6, S. 15). Wie bereits in § 2 gezeigt wurde, kann die kardinale Beschränktheitsbedingung in der Form (4)

$$\prod_{j\in N_k} x_j = 0 \quad \forall\, N_k \subseteqq J \text{ mit } |N_k| > q$$

geschrieben werden.

So ergäbe z.B.

$$|\{x_j > 0 : j\in\{1,2,3,4,5\}\}| \leqq 3$$

die disjunktiven Restriktionen

$$\begin{aligned} x_1\cdot x_2\cdot x_3\cdot x_4\cdot x_5 &= 0 \\ x_1\cdot x_2\cdot x_3\cdot x_4 &= 0 \\ x_1\cdot x_2\cdot x_3\cdot x_5 &= 0 \\ x_1\cdot x_2\cdot x_4\cdot x_5 &= 0 \\ x_1\cdot x_3\cdot x_4\cdot x_5 &= 0 \\ x_2\cdot x_3\cdot x_4\cdot x_5 &= 0. \end{aligned}$$

Es ist aber gar nicht nötig, (4) vor Beginn der Rechnung in der obigen Weise zu spezifizieren. Man modifiziert vielmehr einfach die Definition von "kritisch" (siehe S. 36) wie folgt:

Definition: Bezüglich einer Basislösung x mit der Basis B heiße $t\in J\cap(I_N\setminus I_N^{1*})$ kritisch, wenn $|J\cap(I_N\setminus I_N^{1*})| = |J| - q$ ist. B heiße kritisch, wenn $|J\cap I_B| > q$ ist.

Mit der so abgeänderten Definition wird das Verfahren völlig analog durchgeführt. Beispielsweise wird durch die Regel 1 sichergestellt, daß nur dann Variablen x_t ($t\in J$) in die Basis gelangen, wenn mehr als $|J|-q$ zu J gehörige Nichtbasisvariablen gleich Null und somit weniger als q Variablen x_i ($i\in J$) positiv sind. Als verletzte disjunktive Restriktion $k_o\in K$ (in CUT etc.) wird dabei stets das Produkt ($\neq 0$) sämtlicher positiver zu J gehöriger Variablen aufgefaßt, sobald ihre Anzahl q übersteigt.

10. Implementierung und Rechenerfahrung [1)]

Zur Implementierung des im vorangegangenen Paragraphen beschriebenen Lösungsverfahrens stand vorübergehend das IBM Softwaresystem MPSX/370 zur Verfügung (siehe [IBM, 1979]). Das IBM Mathematical Programming System Extended/370 ist ein IBM Lizenzprogramm welches in erster Linie dazu dient, Probleme der Form (1) - (3) mit Hilfe des in § 3 dargestellten Verfahrens zu lösen. Darüber hinaus bietet es dem Benutzer mittels der Kontrollsprache ECL (Extended Control Language) die Möglichkeit, neue Algorithmen, die auf der Simplexmethode basieren, zu implementieren. Die ECL-Einrichtungen bauen auf der Mehrzweck-Programmiersprache PL/I auf und sind Teile von MPSX/370, es wird jedoch ein PL/I Compiler zur Übersetzung des vom Benutzer in ECL programmierten Kontrollprogramms benötigt. Das übersetzte ECL-Programm wird dann zu PL/I und MPSX/370 Routinen geladen, um einen Modul zu erzeugen, welcher vom Betriebssystem ausgeführt werden kann. Eine derart ausgefeilte Anwendung erfordert jedoch eine überaus detaillierte Kenntnis von MPSX/370, zu deren Erreichung wiederum eine große Anzahl von Versuchsläufen nötig ist. Diese stand in dem hier vorliegenden Fall leider nicht zur Verfügung, so daß einige Schwierigkeiten, die bei der Implementierung des Verfahrens aus § 9 auftraten und insbesondere Abfragen und Veränderungen der bestehenden Arbeitsmatrix betrafen, nicht behoben werden konnten. Aus diesem Grunde konnte nur eine stark vereinfachte Version des in § 9 geschilderten Lösungsverfahrens realisiert werden, welche im wesentlichen durch die folgenden Kürzungen charakterisiert war:

- Die Prozedur REDUCE wurde weggelassen.
- Die Prozedur CUT wurde in Phase 2b und Phase 2c weggelassen.
- In der Prozedur CUT wurde eine neue eingeführte Restriktion unmittelbar in den darauffolgenden Iterationen wieder eliminiert.

---

1) Es sei ausdrücklich betont, daß sämtliche Programmierarbeiten, von denen in diesem Paragraphen die Rede ist, am Rechenzentrum der Ruhrgas AG in Essen von den Herren SCHMIDT und insbesondere RITZENHOFEN durchgeführt wurden.

- Es wurden nur disjunktive Restriktionen der besonders einfachen Gestalt

$$x_i \cdot x_j = 0$$

eingegeben.

Außerdem wurde anstelle der Prozedur BRANCH AND BOUND der Mixed Integer Programming/370 (MIP/370)-Zusatz von MPSX/370 verwendet, welcher dazu dient, gemischt-ganzzahlige Probleme mit Hilfe eines Branch-and-Bound-Verfahrens zu lösen (siehe [IBM, 1975b], [Ciriani, 1980], [Benichou/Gauthier/Hentges/Ribiere, 1977]). Hierfür wurden dann die disjunktiven Restriktionen als lineare Restriktionen mit 0/1-Variablen (siehe S. 14) hinzugefügt.

Das so implementierte Verfahren wurde auf Probleme (1) - (4) [1] verschiedener Größe angewandt. Zum Vergleich wurden diese Probleme von vornherein als gemischt-ganzzahlige 0/1-Optimierungsprobleme formuliert und mit MIP/370 gelöst. Einige typische Resultate sind in der folgenden Tabelle zusammengefaßt. Die implementierte Version des im § 9 beschriebenen Lösungsverfahrens ist dabei mit LODI ("Lineare" Optimierung mit Disjunktionen) abgekürzt.

---

1) Die Transformation auf die Gestalt (1) - (4) ist als Startprozedur in MPSX/370 enthalten.

Problem (1) - (4)	Anzahl der Restriktionen (2)	(3)	(4)	Rechenzeit (Min) MIP	LODI
1	19	35	9	0.05	0.05
2	19	35	9	0.06	0.05
3	19	35	9	0.06	0.06
4	19	35	9	0.09	0.06
5	159	204	15	0.11	0.10
6	52	300	76	0.40	0.18
7	72	600	151	0.72	0.45
8	525	810	60	3.19	1.39
9	132	2000	501	7.21	0.82

Alle Versuche wurden auf einer IBM 3031 durchgeführt. Für LODI ist die beste Rechenzeit eingetragen, die sich bei Variation der Schranken S_i ($1 \leq i \leq 4$) und der Schrittweite $\hat{\lambda}$ ergab. Überraschend ist, daß LODI bei sämtlichen in der Tabelle aufgeführten Problemen (außer bei Problem 8) mit Fall 2 (siehe S. 84) abbrach. Diese Tatsache deutet darauf hin, daß in der Praxis häufig linear optimale Basislösungen existieren, die auch disjunktiv sind, was seine Ursache darin haben dürfte, daß Probleme, die auf praktischen Gegebenheiten beruhen, meist stark entartet sind. Da das Verfahren aus § 9 mit großer Wahrscheinlichkeit in vielen Fällen eine linear optimale und disjunktive Basislösung - sofern sie existiert - auffindet, ohne Phase 2c oder sogar Phase 2b zu durchlaufen, ist der oben erwähnte Umstand ein weiterer großer Vorzug dieser Methode, weil in solchen Fällen dann Rechenzeiten möglich sind, die in etwa denen entsprechen dürften, die sich bei Lösung der zugehörigen linearen Probleme (1) - (3) mit Hilfe des Simplexverfahrens ergeben würden (siehe Problem 9). Der Grund dafür, daß MIP/370 bei den kleineren Problemen gleichschnell war, liegt darin, daß MPSX/370 bei Verwendung eines übergeordneten ECL-Kontrollprogramms (wie in LODI) wesentlich langsamer ist als bei Standardanwendung (wie in MIP/370). Abschließend sei noch erwähnt, daß MPSX/370 eine Startprozedur enthält, die versucht, die Problemgröße von (1) - (3) herabzusetzen, in-

dem (in besonders markanten Fällen) redundante Zeilen und Spalten eliminiert werden, Spalten, deren Grenzen auf Grund anderer Problembeschränkungen nicht übertreten werden dürfen, freigegeben werden und indem bestimmt wird, ob bestimmte Restriktionen in (2) durch Grenzen in (3) ersetzt werden können. Die Wirksamkeit einer solchen Startprozedur läßt sich bei einem Problem der Form (1) - (4) noch wesentlich erhöhen.

Dies sei am Ausgangsproblem von Seite 7 demonstriert.
Setzt man die obere Grenze von $\underline{x}_4$ auf 5, dann kann die Restriktion

$$0 \leqq \underline{x}_1 + \underline{x}_4 \leqq 7$$

gestrichen werden. Weiterhin kann, wegen

$$1 \leqq -\underline{x}_1 + \underline{x}_3 \leqq 3,$$

die untere Grenze von $\underline{x}_3$ auf 1 heraufgesetzt werden, da $\underline{x}_1$ nicht negativ werden darf. Daraus folgt nun aber, wegen

$$(2-\underline{x}_1)\cdot(\underline{x}_3+4) = 0,$$

daß $\underline{x}_1$ auf seine obere Grenze fixiert werden kann, und daraus folgt wiederum, wegen

$$\underline{x}_1\cdot(\underline{x}_3+4)\cdot(4-\underline{x}_3) = 0,$$

daß auch $\underline{x}_3$ auf seine obere Grenze fixiert werden kann etc.
Eine disjunktive Restriktion der Form

$$\underline{x}_1\cdot\underline{x}_3 = 0$$

würde demnach bereits vor Beginn der eigentlichen Rechnung die Unlösbarkeit des Problems anzeigen.

Schlußbemerkung

Eine Erweiterung von MPSX/370 könnte das folgende Aussehen haben:

- Einbeziehung der Zusatzregeln in die Standardpivotauswahl,
- Aufnahme von CUT, BRANCH AND BOUND und REDUCE als Optimierungs- bzw. Analysierungsprozeduren,
- Verfeinerung der Startprozedur zur Herabsetzung der Problemgröße durch Ausnutzung der in den disjunktiven Restriktionen enthaltenen Informationen.

Ein derartiges Softwaresystem würde ein wirkungsvolles Zusammenspiel simplexartiger Verfahren mit Lösungsprinzipien der ganzzahligen Optimierung (wie in §9 beschrieben) garantieren und gleichzeitig das Ergebnis einer jahrelangen Entwicklung der Auslegung, Implementierung und Verwendung von linearen Optimierungssystemen beinhalten. Die Klasse der effizient lösbaren Probleme würde dadurch bis in die nichtlineare Optimierung hinein vergrößert werden.

Literaturverzeichnis

Balas, E. [1977] Disjunctive Programming. Management Science Research Report No. 415, Carnegie-Mellon University.

Bastian, M. [1976] Lineare Komplementärsprobleme im Operations Research und in der Wirtschaftstheorie. Verlag Anton Hain, Meisenheim am Glan.

Benichou, M.; J.M. Gauthier; G. Hentges; G. Ribiere [1977] The Efficient Solution of Large Scale Linear Programming Problems - Some Algorithmic Techniques and Computational Results. Mathematical Programming 13, 280 - 322.

Brauer, K.M. [1969] Binäre Optimierung. Carl Heymanns Verlag KG, Köln-Berlin-Bonn-München.

Ciriani, T.A. [1972] The Free Parametrization for Linear Programming with Bounded Variables. Rendiconti del Seminario Matematico della Universita di Padova, Vol. 47, 77 - 107.

Ciriani, T.A. [1980] Some Optimization Features of MPSX/370. International Workshop on Advances in Linear Optimization Algorithms and Software, Pisa, Italy.

Collatz, L.; W. Wetterling [1966] Optimierungsaufgaben. Springer Verlag, Berlin-Heidelberg-New York.

Courant, R [1963] Vorlesungen über Differential- und Integralrechnung. Zweiter Band, Springer Verlag, Berlin-Göttingen-Heidelberg.

Driebeek, N.J. [1966] An Algorithm for the Solution of Mixed Integer Programming Problems. Management Science 12, 576 - 587.

Geoffrion, A.M.; R.E. Marsten [1972] Integer Programming Algorithms: A Framework and State-of-the-Art-Survey. Management Science 18 (9), 465 - 491.

Glover, F. [1973] Convexity Cuts and Cut Search. Operations Research 21, 123 - 134.

Hadley, G. [1962] Linear Programming. Addison-Wesley Publishing Company.

Hadley, G. [1969] Nichtlineare und dynamische Programmierung. Physica-Verlag, Würzburg-Wien.

Henn, R.; H.P. Künzi [1968] Einführung in die Unternehmensforschung II. Springer-Verlag, Berlin-Heidelberg-New York.

Ibaraki, T. [1971] Complementary Programming. Operations Research 19 (6), 1523 - 1529.

Ibaraki, T. [1973] The Use of Cuts in Complementary Programming. Operations Research 21, 353 - 359.

IBM Corporation [1975a] An Introduction to Modelling Using Mixed Integer Programming. Application Description Manual.

IBM Corporation [1975b] IBM Mixed Integer Programming/370 (MIP/370). Program Reference Manual, SH 19-1099 VII.

IBM Corporation [1979] IBM Mathematical Programming System Extended/370 (MPSX/370). Program Reference Manual, SH 19-1095 IX.

Jeroslow, R.G. [1977] Cutting Plane Theory: Disjunctive Methods. Annals of Discrete Mathematics 1, 293 - 330.

Kall, P. [1976] Mathematische Methoden des Operations Research. Teubner Studienbücher.

Kaplan, S. [1971] Comment on a Precis by Shanno and Weil. Management Science 17 (11), 778 - 780.

Neumann, K. [1975] Operations Research Verfahren. Band 1, Carl Hanser Verlag, München-Wien.

Orchard-Hays, W. [1968] Advanced Linear Programming Computing Techniques. McGraw Hill, New York.

Owen, G. [1973] Cutting Planes for Programs with Disjunctive Constraints. Journal of Optimization Theory and Applications 11, 49 - 55.

Rammé, M. [1975] Optimierung allgemeiner quadratischer Programmierungsprobleme. Dissertation, Hamburg.

Ravindran, A.; T.W. Hill [1973] A Comment on the Use of Simplex Method for Absolute Value Problems. Management Science 19 (5), 581 - 582.

Shanno, D.R.; R.L. Weil [1971] "Linear" Programming with Absolute Value Functions. Operations Research 19 (1), 120 - 124.

Simonnard, M. [1966] Linear Programming. Prentice Hall Inc., Englewood Cliffs, N.J.

Telgen, J. [1979] Redundancy and Linear Programs. Doctoral Thesis, Economische Wetenschappen aan de ERASMUS Universiteit, Rotterdam, Netherlands.

Thompson, G.L.; F.M. Tonge; S. Zionts [1966] Techniques for Removing Nonbinding Constraints and Extraneous Variables from Linear Programming Problems. Management Science 12 (7), 588 - 608.

Winkels, H.-M. [1979] Die Bestimmung aller Lösungen eines linearen Restriktionssystems und Anwendungen. Arbeitsbericht Nr. 15 des Instituts für Unternehmungsführung und Unternehmensforschung der Ruhr-Universität Bochum, D-4630 Bochum.

Young, R.D. [1971] Hypercylindrically Produced Cuts in Zero-One Integer Programs. Operations Research 19 (6), 1393 - 1405.

Zoutendijk, G. [1976] Mathematical Programming Methods. North-Holland Publishing Company, Amsterdam-New York-Oxford.

Lebenslauf

Name:	Hans Roland Weinberger
Adresse:	Am Erlenkamp 9, 4630 Bochum
Telefon:	0234 / 70 10 32
Geburtstag:	15. 01. 1951
Geburtsort:	Nördlingen (Bayern)
Familienstand:	Verheiratet, 1 Kind
Schulbildung:	
1957-1961	Volksschule in Wamel/Möhnesee
1961-1964	Aldegrever-Gymnasium in Soest
1964-1970	Friedrich-Spee-Gymnasium in Geldern
15.06.1970	Abitur
Wehrdienst:	
1970-1971	Ausbildung zum Stabsdienstsoldaten
Studium:	
1971-1977	11 Semester Mathematik mit Nebenfach Wirtschaftswissenschaft an der Ruhr-Universität Bochum
26.04.1977	Diplom
1977-1981	Anfertigung der Dissertation
Berufstätigkeit:	
1974-1975	Studentische Hilfskraft
1977-1977	Wissenschaftliche Hilfskraft
1977-1981	Wissenschaftlicher Assistent (m.d.V.b.) an der wirtschaftswissenschaftlichen Abteilung der Ruhr-Universität Bochum (Lehrstuhl Prof. Dr. Arno Jaeger)